(continued from Part 7)

Transformers

Electronic circuits need supply voltages that are different from the 'mains'. Mains voltage can be changed to almost any other level by means of a **transformer**.

A transformer consists of two adjacent coils of wire, which are wound onto the same metal core, sometimes known as a former. They have no direct electrical connection between them. When a voltage is applied to one of the coils – which is called the **primary** winding – an electric field is produced around it. If this voltage alternates, the electric field will fluctuate with the voltage. As a result a similarly fluctuating voltage is induced in the other – **secondary** – coil. The size of this **induced voltage** is proportional to the ratio of the number of turns of wire in each winding.

To obtain a lower voltage than is available, a **step-down** transformer is used. This has fewer turns in the secondary winding than in the primary. Higher voltages than the input can be obtained by using a **step-up** transformer. This has more turns in the secondary winding than in the primary.

The voltage obtained from the secondary winding of a transformer is:

$$V_s = \frac{N_s}{N_p} \times V_p$$

where:

V_s = induced secondary voltage
V_p = applied primary voltage
N_s = number of turns in secondary coil
N_p = number of turns in primary coil

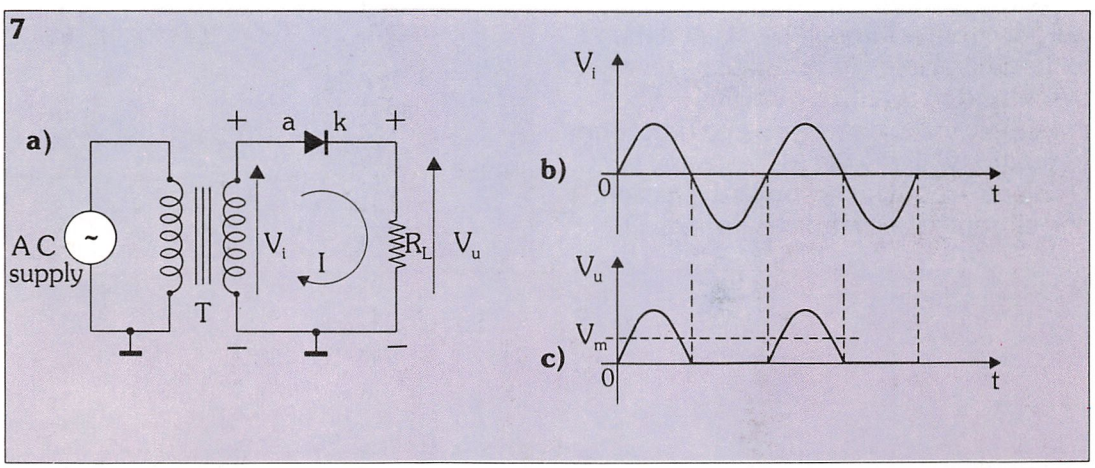
Rectifier circuits

The simplest power supply circuit is the half-wave rectifier which uses a single diode (*see figure 7a*).

Here, a transformer is connected to the mains supply to transform the mains voltage (240 V) to the level required by the circuit (V_i).

The alternating voltage in the secondary coil, V_i, is applied to the diode which conducts only when the anode voltage is greater than the cathode voltage. If the voltage V_i is high enough the presence of the diode's threshold voltage V_{th} can be discounted.

As we know, only the positive part of the input signal can pass through the diode and reach the output terminals. The result of this is that a series of current pulses flow when V_i is positive. These pulses all flow in the same direction, but their magnitude is not constant. These give a similar voltage V_u across the load resistor R_L.



7.a) Half-wave voltage rectifier circuit.
b) shows the wave form on the secondary winding of the transformer.
c) shows the wave form on the load.

Figure 7b shows a sinusoidal input voltage wave form, such as that used in the mains supply where the voltages alternates about 0 V at a frequency of 50 Hz. As you can see, the positive half cycles are like the negative half cycles, so the average value is 0 V. On the other hand *figure 7c* shows the half-wave rectified output voltage, whose average or mean value, V_m, is clearly greater than 0 V.

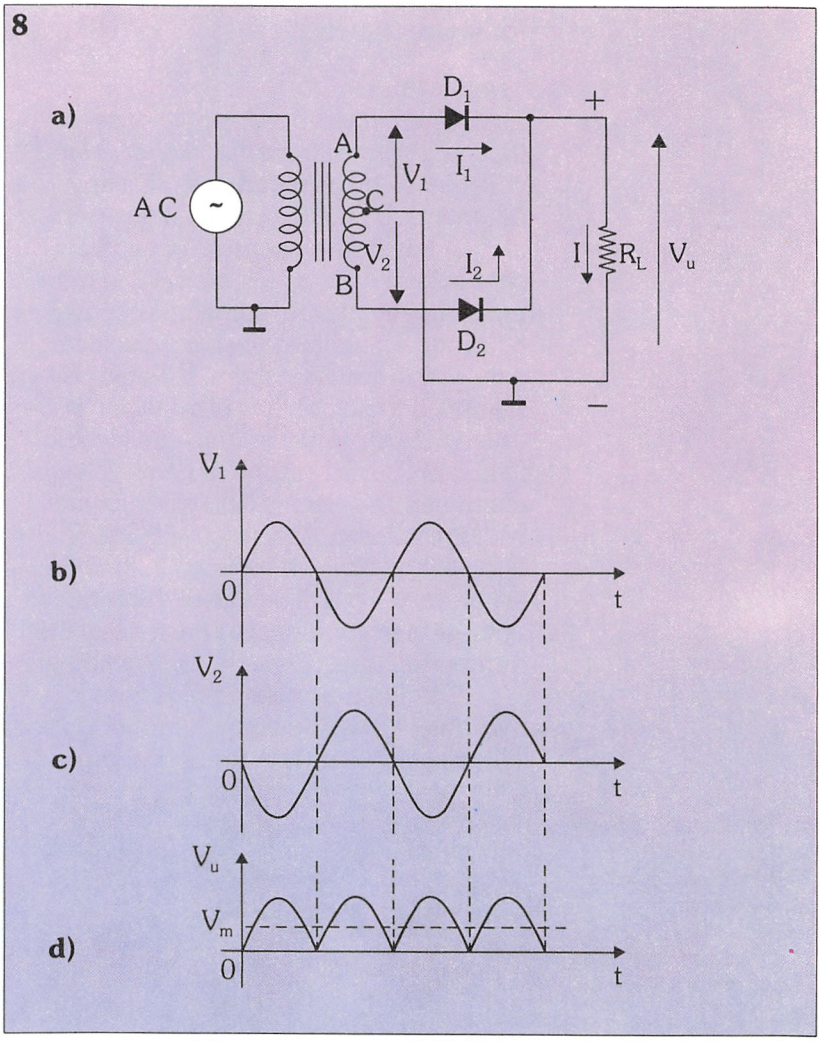
Full wave rectifiers

The half wave rectifier gives a very variable output voltage which is useless in most cases; the current has continuous changes

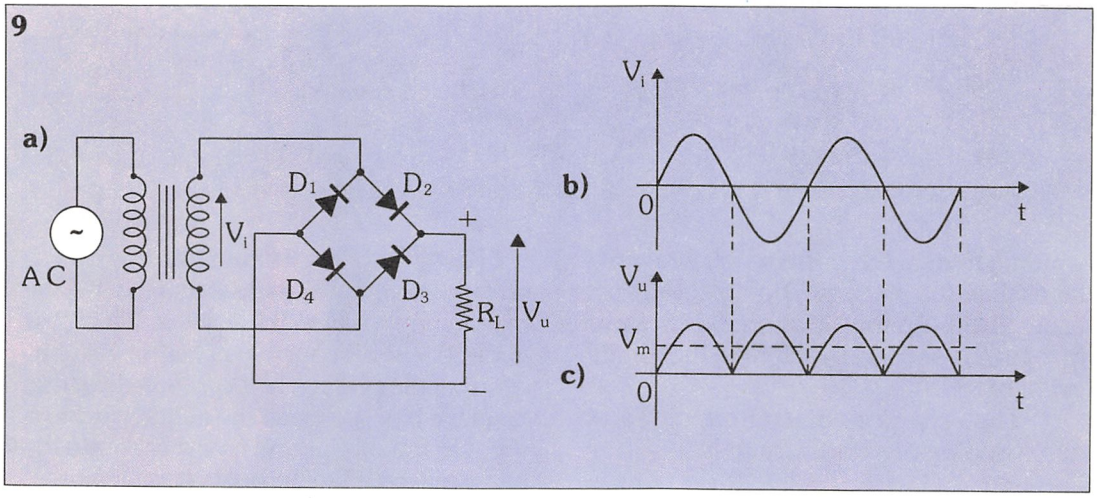
of amplitude and since half of the input voltage is unused, the efficiency of the half-wave rectifier is low. There are two ways of solving this problem and increasing the efficiency of rectification. The first is to use a **full wave rectifier,** which has two diodes and ulitizes both halves of the wave. A full wave rectifier (*figure 8a*) can be considered as two single half wave rectifiers.

Here the secondary transformer is divided into two parts by a central tap. The two resulting voltages V_1 and V_2 are half a wave out of phase with each other. In this way the two diodes in the circuit conduct alternately, each for a half wave. Consequently the average V_m at the output will be double that of the half wave rectifier. *Figures 8b and 8c* show the wave forms V_1 and V_2 out of phase from each other.

Let's see exactly what happens when a voltage is applied to this circuit. When A is positive with respect to the central tap C, B will be negative with respect to C. Diode D_1 will be forward biased and so current will flow from A through D_1, R_L and round to the central tap; diode D_2 will be reverse biased and so no current will flow. When the supply voltage reverses, and B becomes positive with respect to the central tap C, D_2 is now biased in the forward direction and current will flow from B through D_2 and R_L to the centre tap. The resultant wave form is shown in *figure 8d,* and is made up of the positive halves of V_1 and V_2. Thus current flows to the load during both half cycles.



8.a) Full wave voltage rectifier. b) and c) show the wave forms on the two halves of the transformer secondary coil. d) shows the wave form on the load.



9.a) Bridge rectifier circuit. b) shows the waveform on the secondary winder of the transformer. c) shows the waveform present on the load.

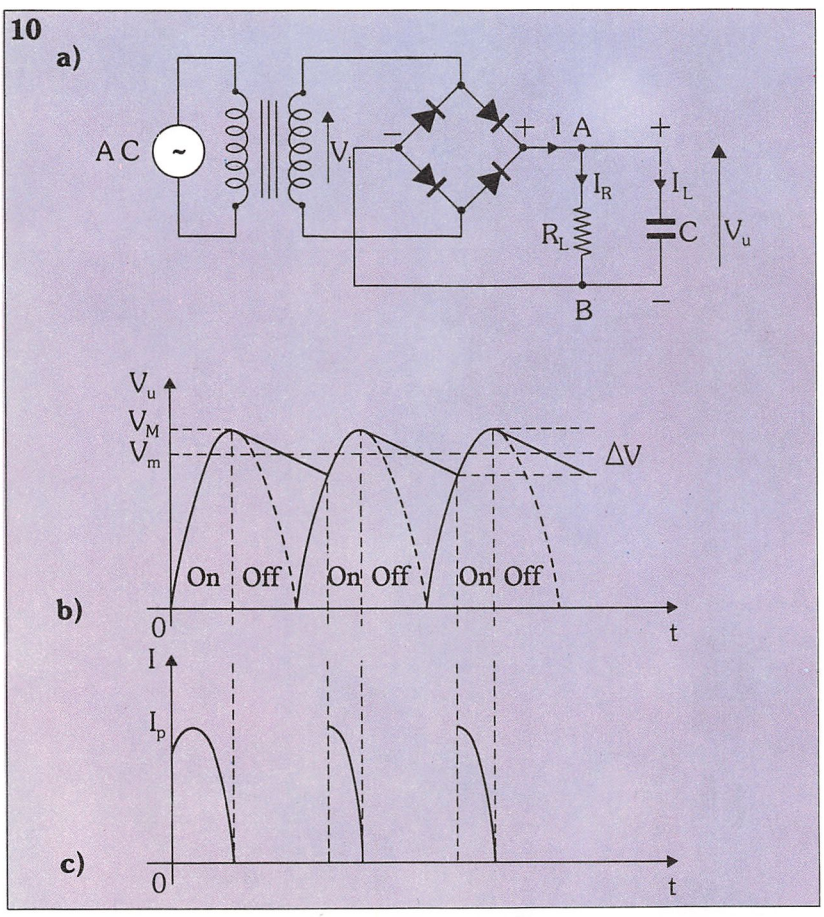
One disadvantage of using this circuit is that the transformer's secondary coil must produce twice the voltage of that used in the half-wave rectifying circuit because only half of the secondary winding is used at any one time. So great care must be taken with this circuit to check that the

reverse voltage applied to the two diodes does not exceed their maximum value.

The second method of increasing efficiency is the **full wave bridge rectifier,** illustrated in *figure 9a.* This overcomes the problem of the high reverse voltage as no central tap is needed. The circuit therefore uses half the voltage of the previous rectifier.

Here four diodes are used as shown. When the input voltage from the transformer is positive, i.e. the voltage is in the direction of the arrow V_i in the diagram, diodes D_2 and D_4 conduct – that is, the current takes a route through D_2, along R_L, through D_4 and back to the secondary terminal. When the voltage is negative, D_3 and D_1 conduct, while the others are off.

The resulting wave is shown in *figure 9c.* As you can see, the average value of the output is the same as for the previous full-wave rectifier but only half the input voltage is required. So the reverse voltage to which the diodes are subjected is much less than in the previous configuration.



10.a) Bridge rectifier with a capacitive filter. b) shows the waveform present on the load. c) shows the waveform of the current flowing in the bridge.

Smoothing circuits

The output voltages of the circuits examined so far are pulsating, undirectional voltages. They do not satisfy the need for steady, continuous voltage. So it is absolutely necessary to use **smoothing circuits** to level out the wave form at the rectifier's output. Smoothing out the waveform also effectively raises the average voltage.

The simplest smoothing filter is a large **capacitive filter** connected as in *figure 10a.* We have here a combination of bridge, load resistance and filter capacitor, with the capacitor in parallel with the load.

During the on phase of the diode bridge the output current charges the smoothing capacitor to a maximum voltage which is the output voltage, V_m. When the input voltage starts to diminish the voltage on the capacitor stops the bridge from working because the capacitor voltage is greater than that of the diode's anode. The capacitor then discharges through R_L and in doing so the voltage of the capacitor is reduced slightly.

In the following half wave the bridge will begin to conduct again only when the input voltage V_i exceeds the voltage on the capacitor. At that moment the capacitor will recharge until it reaches its maximum voltage, when it takes over again. The resultant wave form is shown in *figure 10b.*

The current in the bridge will only circulate when the capacitor is charging, so the diode current can become quite high. This is a fact to bear in mind when choosing the type of bridge to be used. The average voltage across the load resistor almost reaches the maximum supply voltage.

A small ripple still remains in the wave and this variable component is called the **ripple voltage, Δ V.** Δ V depends on the size of the capacitor used; the higher the capacitance the lower the ripple voltage. For this reason very big capacitors are used – in the order of tens to thousands of microfarads. However, it is important not to choose a capacitor too large for the circuit, as it might cause current surges which are too high for the bridge to handle. Another point to consider is that the slower the rate of discharge from the capacitor, the shorter the time available to recharge.

The capacitors used for smoothing power supplies are usually electrolytic types. As these are polarized, the '+' terminal must be connected to the positive side of the rectifier circuit.

Smoothing circuits using inductors

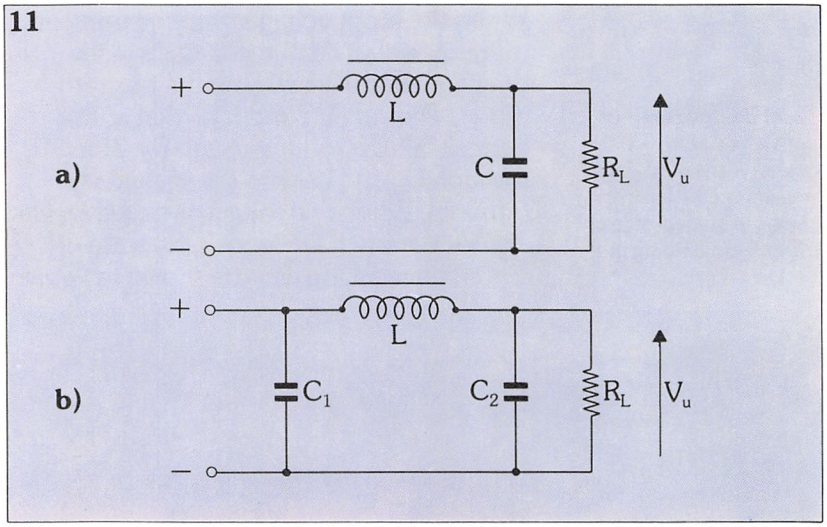
There are other types of smoothing filters which use a combination of inductors and capacitors – these are known as **LC filters.** The letter L is the traditional abbreviation for an inductor. Two typical combinations are given in *figure 11*.

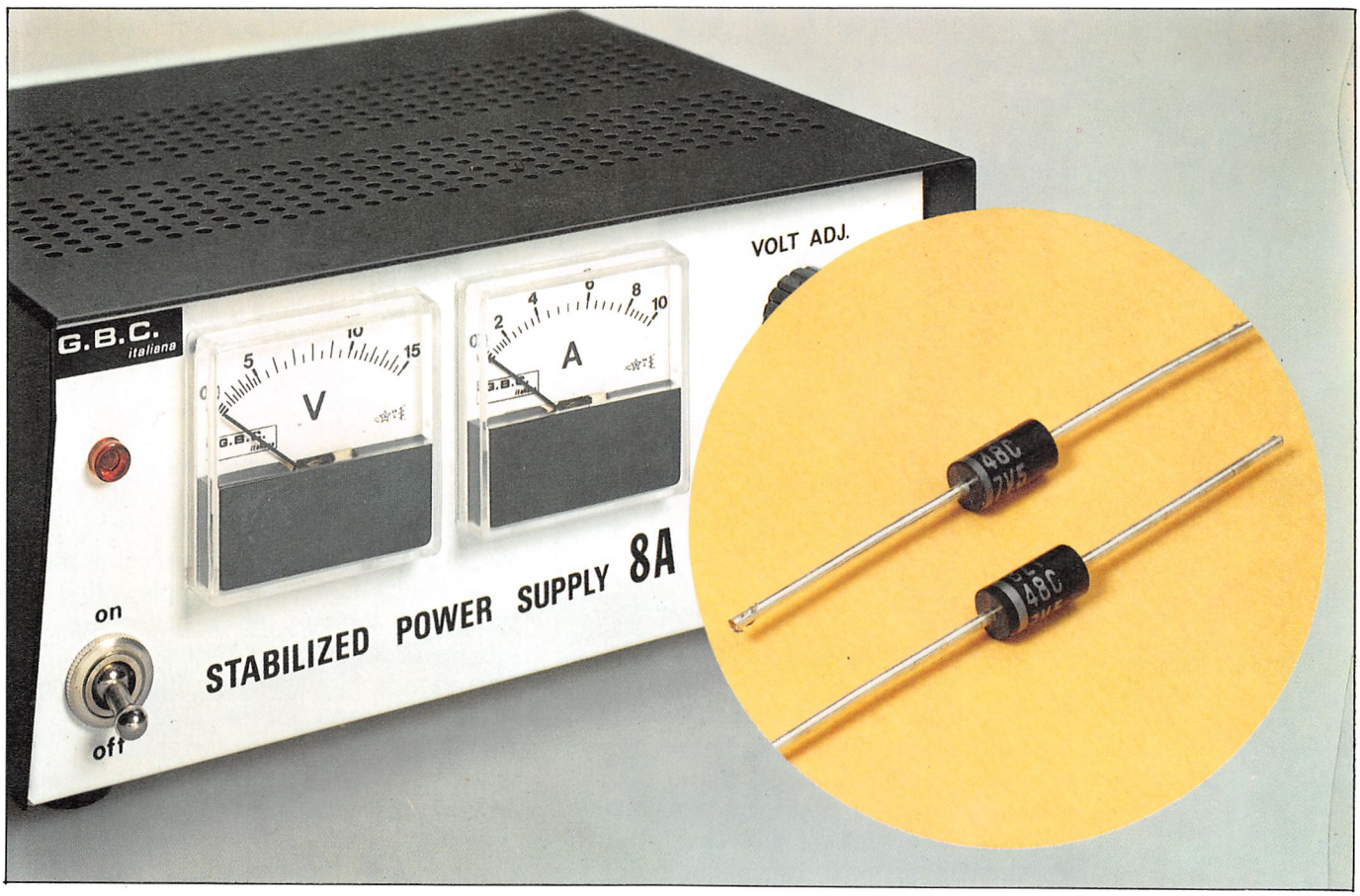
The **inductor coil** used is known as a **choke**. This helps to smooth out the wave by storing up energy in a magnetic field when the current through it tends to increase. This stored energy is converted back into current flow whenever the voltage across its terminals tends to fall.

If the first component closest to the rectifier is a choke, as in *figure 11a,* the circuit is known as a **choke-input filter,** but if the first component is a capacitor as in *figure 11b*, the circuit is a **capacitor-input filter**.

These filters are in general more expensive than ones which use a single capacitor, but they do guarantee a substantial reduction in the ripple voltage. Another drawback to these LC filters is that they supply a voltage lower than that supplied by a single capacitor filter, because of the voltage absorbed by the extra circuitry.

11. LC filters are smoothing circuits which use a combination of inductors and capacitors. a) shows a choke-input filter, where the choke is the component closest to the rectifier. b) is a capacitor-input filter, where the capacitor is the first component.





Stabilized supplies

The supplies looked at so far are very simple but they have three fundamental drawbacks. First, the output voltage never remains constant but varies with the variation in the load. Second, the continuous output voltage varies with the alternating input voltage. Frequently the mains can vary by as much as 10%, while the voltage required by the circuit must usually remain constant. Third, the output voltage varies with the temperature because of the behaviour of semiconductor devices.

There are several types of circuits, called **stabilized supplies,** which are built to solve these problems.

In general these circuits use full wave or bridge rectifiers to rectify the alternating voltage. The rectified voltage is then smoothed, using a big capacitor which greatly reduces the ripple voltage. The real stabilizing stage follows the capacitor.

A voltage stabilizing IC is almost always used for this stage. Such ICs guarantee a constant output voltage, accurate within a few per cent, even with variations in load current, temperature and input voltage–in fact they come close to producing ideal power supplies for current requirements of anything up to about 5 A.

If, however, voltage stabilization is required for load currents up to about 50 mA, then a Zener diode can be used, the characteristics of which have already been mentioned in an earlier chapter.

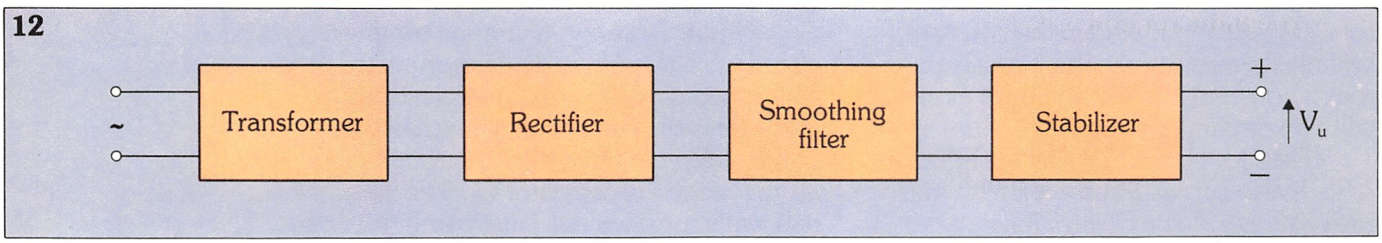
The Zener diode used as a stabilizer

Figure 12 shows a schematic diagram of a stabilized supply. We will only examine the stabilizing block as all the others have been covered in the preceding paragraphs. *Figure 13* is the circuit diagram of a voltage stabilizer using a Zener diode. The output voltage is taken parallel to the Zener diode i.e. it is the breakdown voltage of the diode, while there is a protection resistor R_p placed between the input voltage and the Zener diode.

We'll now see how this circuit can maintain a constant output voltage with the variation of both the input voltage and the current absorbed by the load.

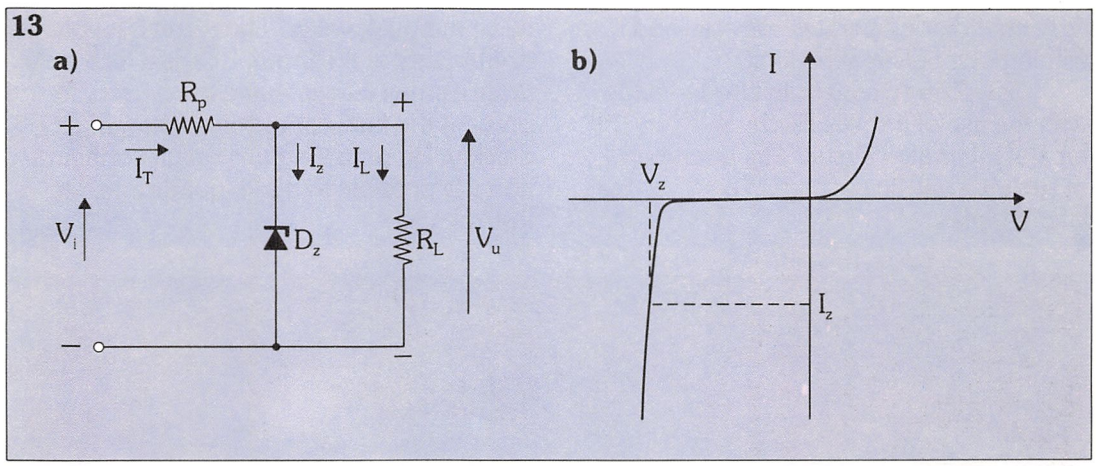
You will remember that when reverse biased, a Zener diode maintains a constant voltage, even when the drawn current varies. When the input voltage increases, with an increase in the total current I_T, the drop in voltage across R_p increases as does the current absorbed by the Zener. In this way the current I_L and the voltage V_u remain constant.

12



12. Schematic diagram of a stabilized supply.

13



13. a) Circuit diagram of a voltage stabilizer using a Zener diode. b) shows the IV curve for a Zener diode.

Stabilizer power supply and (inset) two Zener diodes, which are used in the stabilizing circuit. (Photo: GBC)

However, when the input voltage V_i diminishes so does the voltage across R_p. Consequently the current I_z will also reduce, compensating for the voltage variation in V_i.

So what happens if the current required by the load increases and the current I_T supplied by the previous stages stays the same? The current I_z through the Zener diode will diminish while the voltage across it remains constant.

In the same way, when the current absorbed by the load is reduced, the Zener diode will absorb a greater amount of the total current and maintain the constant voltage. In these situations, since I_T remains constant, the voltage drop across R_p will also remain constant.

Resistor R_p drops a voltage equal to $I_T R_p$. It also acts as a current limiter and prevents too high a current flowing through the Zener diode. This ensures that the Zener diode operates correctly and within power dissipation limits.

Stabilizing circuits can also be made with the use of a specialised type of integrated circuit. The operation of such ICs will be covered in a later chapter.

Voltage multipliers

Sometimes very high voltages are needed, such as the **extra high tension (EHT)** used in colour televisions. Here the circuits we have looked at so far would be impractical and too expensive because of the very large size of transformer needed. **Multiplying circuits** are used instead. These allow high voltages to be obtained from the ordinary mains supply. The only limitation of these circuits is that any load connected to the voltage multiplier must absorb a very small current.

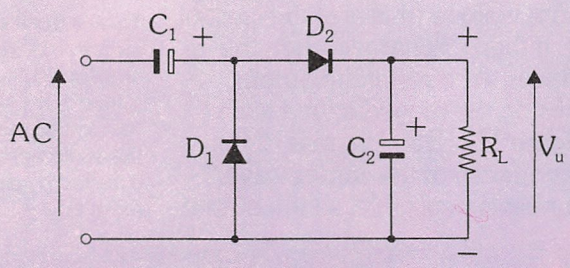
Figure 14 shows a **voltage doubler** circuit which uses two diodes connected in series, two capacitors (usually electrolytic) and a resistor. The first negative half wave of the alternating signal goes through diode D_1 to charge the capacitor C_1. The second half wave, positive this time, combines with the voltage on C_1 to charge C_2, via D_2, to twice the peak value. The result is a voltage double that at the input. Loads which only draw a small current have to be used, to prevent the final capacitor discharging too quickly. By repeating this circuit configuration it is possible to triple, quadruple, etc. the voltage.

14. A voltage doubler circuit using two diodes in series, two capacitors and a transistor.

Left: Stabilized power supply. Alternating current from the mains is converted into direct current.

Right: a modern radio transmitter/receiver. Another important application of diodes is in peak detector circuits, which are used in all AM radio receivers.

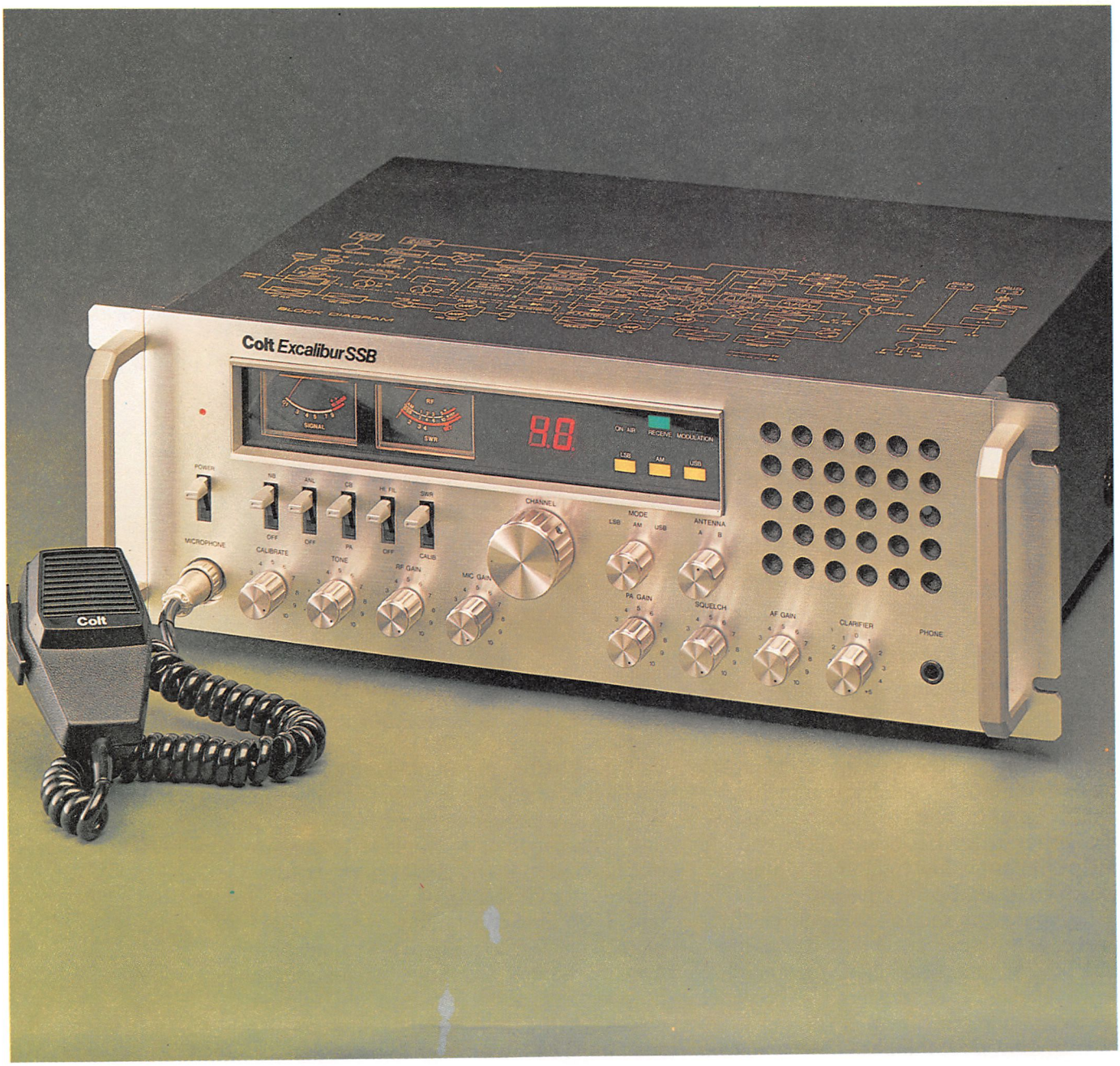
14



Peak detectors

Another important use of the diode is in a **peak detector** circuit like the one shown in *figure 15a.* It is used in all AM radio receivers.

Amplitude modulation has already been mentioned in an earlier chapter and will be covered in greater detail in the later section on communications. You will remember that in amplitude modulation broadcasting, the signal which contains the information is generally a low frequency



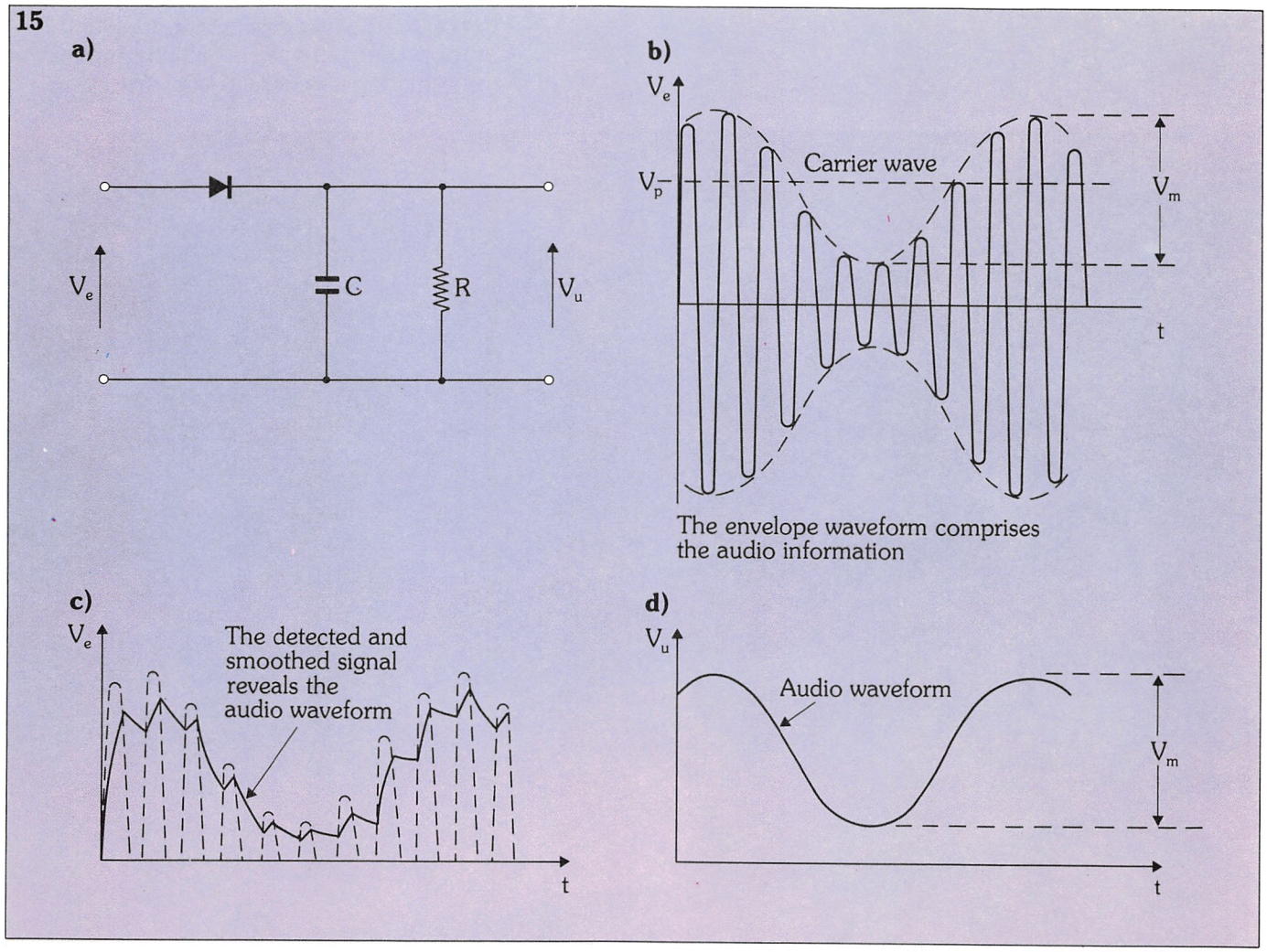
(50 – 20,000 Hz) audio waveform. For this information to be transmitted a very high frequency electrical waveform (in the order of megahertz) is needed. Transmission is achieved by using an oscillator and an amplitude modulator. This high frequency waveform is called the **carrier** because it is modulated to carry the information signal.

At the receiver end the reverse operation must be carried out – the signal must be **demodulated** – which is where the diode comes in. This operation eliminates the carrier, which has done its job, and reconstitutes the low frequency signal which contains the information.

Figure 15b shows the signal which arrives at the receiver. The diode has the task of eliminating the negative part of the wave received as shown in *15c*. The signal received is the one drawn in *15d*.

So how does the peak detector circuit achieve this? When the diode is conducting the capacitor C becomes charged towards the maximum voltage (that is at the crest of the waves). It then discharges when the voltage reduces. As a result the output voltage takes on the ragged edged form shown in *figure 15c*. But, because of the very high frequency of the carrier wave, this unevenness is negligible, so the output voltage is better represented by *figure 15d* – which is, of course, a replica of the original low frequency signal.

This can then be amplified and reproduced as sound. The choice of capacitor and resistor for this circuit is very important. Their time constant ($\tau = RC$) must be high enough to prevent the capacitor from discharging too much between successive peaks of the high frequency carrier waves. On the other hand the time constant must be low enough to allow the



15.a) a peak detector circuit. b) shows the amplitude modulated signal, c) the signal after it has passed the diode, d) the audio signal as detected by an AM radio receiver.

voltage on the capacitor to follow the wave form of the audio signal. This circuit is also called an **envelope detector** because it gives the envelope of all the input signals at the output.

Other circuits using diodes

There are many other circuits which use diodes besides those which we have just looked at.

Just think of the circuits necessary for the functioning of calculators; they often need an electronic device with two functioning states. A diode can carry out this function precisely because it can be either in the ON or OFF state, and can pass from one to the other very simply. Another common use is in a **clamping circuit,** to clamp a wave form to a preset level.

However it is not necessary to cover each of these uses in detail, as you now know enough about the workings of diodes to understand their different functions.

Glossary

capacitive filter	a capacitor, commonly used to smooth or level out rectified voltages in power supplies
diode characteristic	a graph of the current through a diode against the voltage across it. Useful when comparing diodes
full wave rectifier	a circuit that converts every half cycle of an alternating current into direct current
half wave rectifier	a circuit that converts the positive (or the negative) half cycles of an alternating current into direct current
limiting circuit	a circuit which limits waveforms and allows only signals which exceed a preset reference level to pass through the circuit
peak detection	demodulation of a low frequency informational signal, previously modulated on a high frequency AM carrier
power supplies	any source of electrical power in a form suitable for operating electronic circuits
ripple voltage	the ripple which remains after smoothing a rectified voltage with a capacitor. The ripple voltage ($\triangle V$) depends on the size of the capacitor used
stabilized supplies	these give a constant output voltage and current, even with variations in the load, temperature and input voltage. Integrated circuits are usually employed for this purpose, but Zener diodes are occasionally used
voltage multiplier	an arrangement of diodes which allow high voltages to be obtained from the supply network. Depending on the arrangement, voltages can be doubled, trebled, quadrupled, etc.

BASIC THEORY REFRESHER

ELECTRICAL TECHNOLOGY

Using capacitors

Capacitors can be connected within a circuit either in series or in parallel. For the purpose of circuit design, it's important to be able to calculate the total capacitance involved in either type of connection.

Series connection

Figure 1a shows capacitors connected in series. When a voltage is applied across the whole circuit, each capacitor becomes equally charged – whatever their values. This makes sense when you think about it, because any charge which builds up on one capacitor plate causes an equal but opposite charge on the other plate. This in turn induces a charge equal to the first on the next plate of the next capacitor, and so on.

Let's call the charge in each capacitor Q when a voltage V is applied to the capacitors in the series, and call the capacitance or size of each capacitor C_1, C_2, C_3 etc. The whole series of capacitors will itself have a capacitance C, and the voltage drop across each capacitor V_1, V_2, V_3 etc.

The formula for relating the charge, voltage drop and capacitance for each capacitor is:

$$V_1 = \frac{Q}{C_1} \quad V_2 = \frac{Q}{C_2} \quad V_3 = \frac{Q}{C_3} \text{ and so on.}$$

The bigger the capacitor, the smaller the voltage drop. We also have the formula for the whole circuit:

$$V = \frac{Q}{C}$$

As we know that $V = V_1 + V_2 + V_3$ we have:

$$\frac{Q}{C} = \frac{Q}{C_1} + \frac{Q}{C_2} + \frac{Q}{C_3}$$

Dividing both sides by Q we obtain:

$$\frac{1}{C} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} \ldots\ldots$$

In fact, the formula for capacitors in series is of the same form as that for resistors in parallel. If there are only two capacitors in series then:

$$C = \frac{C_1 \times C_2}{C_1 + C_2}$$

Capacitors in parallel

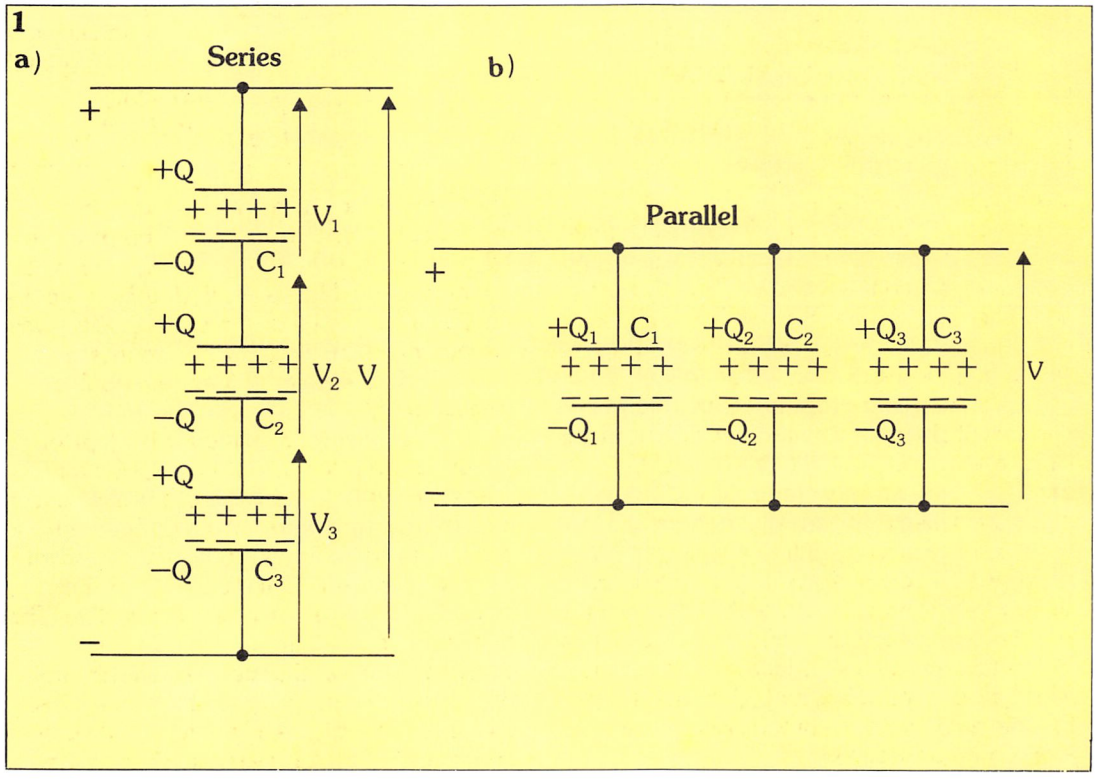
Capacitors in parallel, *see figure 1b*, are arranged so that the same voltage is applied to all of them. In this case the charge on each capacitor will be dependent on its size or capacitance, ie:

$Q_1 = C_1 \times V$, $Q_2 = C_2 \times V$, $Q_3 = C_3 \times V$ etc.

For the whole circuit $Q = C \times V$.

Since $Q = Q_1 + Q_2 + Q_3$

then $C = C_1 + C_2 + C_3$



1. Capacitors connected in series and in parallel.

So capacitors connected in parallel add up like resistors connected in series.

Table 1 compares the connection of capacitors, in series and in parallel. From this you can see that capacitors are linked in parallel if a capacitance increase is required while maintaining a constant voltage. However, capacitors are linked in series when it's necessary to have a circuit whose voltage is greater than that which could be tolerated by a single capacitor.

There are many types of capacitors and they are classified according to their uses, materials and construction. These can be **fixed** or **variable,** according to whether their capacitance is constant or can be modified at will. They can also be distinguished by the type of dielectric used, which may be solid, gaseous (air) or composed of a film of oxide.

Variable capacitors are made up of two systems of alternating plates, one fixed and the other able to rotate, so that the area of the facing surface can be varied at will; in these cases the dielectric is nearly always air.

There are many different types of fixed capacitors, made with a variety of dielectrics. **Mica capacitors** are made with leaves of mica, of suitable thickness and size, which are fitted in between metal plates, usually made of silver. This type of capacitor is for high frequency use in radio engineering or broadcasting, or for high power applications.

Ceramic capacitors, as their name suggests, use a ceramic dielectric. These have a very high dielectric constant, which means that they can be used to construct very small components of very high capacitance.

They can be manufactured in many different shapes and forms, and are chiefly used in telecommunications circuits – particularly where space is at a premium.

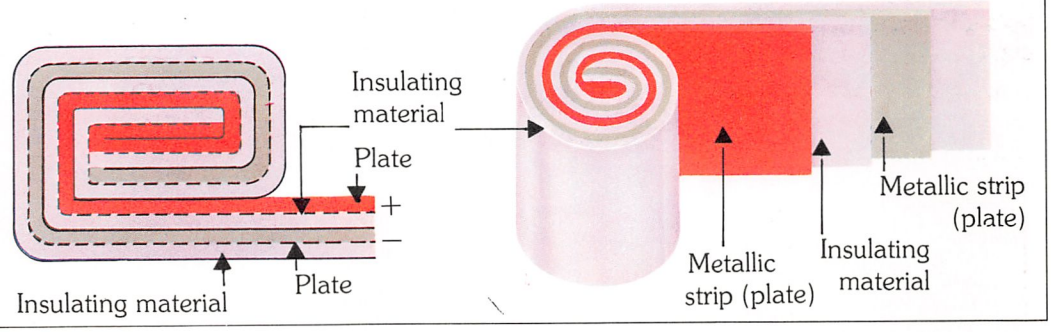
Wrapped capacitors are made up of long metallic strips, such as sheets of aluminium, sandwiched between paper or plastic dielectrics. These are wrapped in a cylindrical or rectangular form. This produces a capacitor with a large useful surface area in a compact body. Moreover, during the wrapping process a second dielectric is wrapped between the outer surfaces of the plates, so that the useful surface and capacitance are doubled (*see figure 2*).

An **electrolytic capacitor** is a type of wrapped capacitor which can be polarized. It is made by placing an aluminium sheet next to a piece of paper or gauze which is saturated in an electrolytic paste. Here, the aluminium sheet is effectively the anode and the gauze is the cathode – as the electrolyte is conductive. The first flow of current through the capacitor causes a chemical change in the aluminium, depositing an oxide layer on the sheet. This creates a dielectric, as aluminium oxide is not conductive. The polarized capacitor can only

Table 1
Series and parallel connection of capacitors

Type of connection	Equivalent capacitance	Charge	Voltage
Parallel	Greater than any individual capacitance. The sum of the capacitances	Proportional to the capacitance of each element	The voltage across each capacitor is equal to the input voltage of the circuit
Series	Smaller than that of the smallest capacitor. The sum of the reciprocal of each capacitance	The same for each capacitor	In inverse proportion to the supply voltage

2



2. Structure of a wrapped capacitor. Long metallic strips are interleaved with layers of an insulating material such as paper or plastic.

be used for direct current applications, but it has the advantage of small size – the non-polarized type takes twice the space to give the same capacitance.

Electrolytic capacitors are used to smooth continuously varying currents, to form links with transistors, to start engines, and so on.

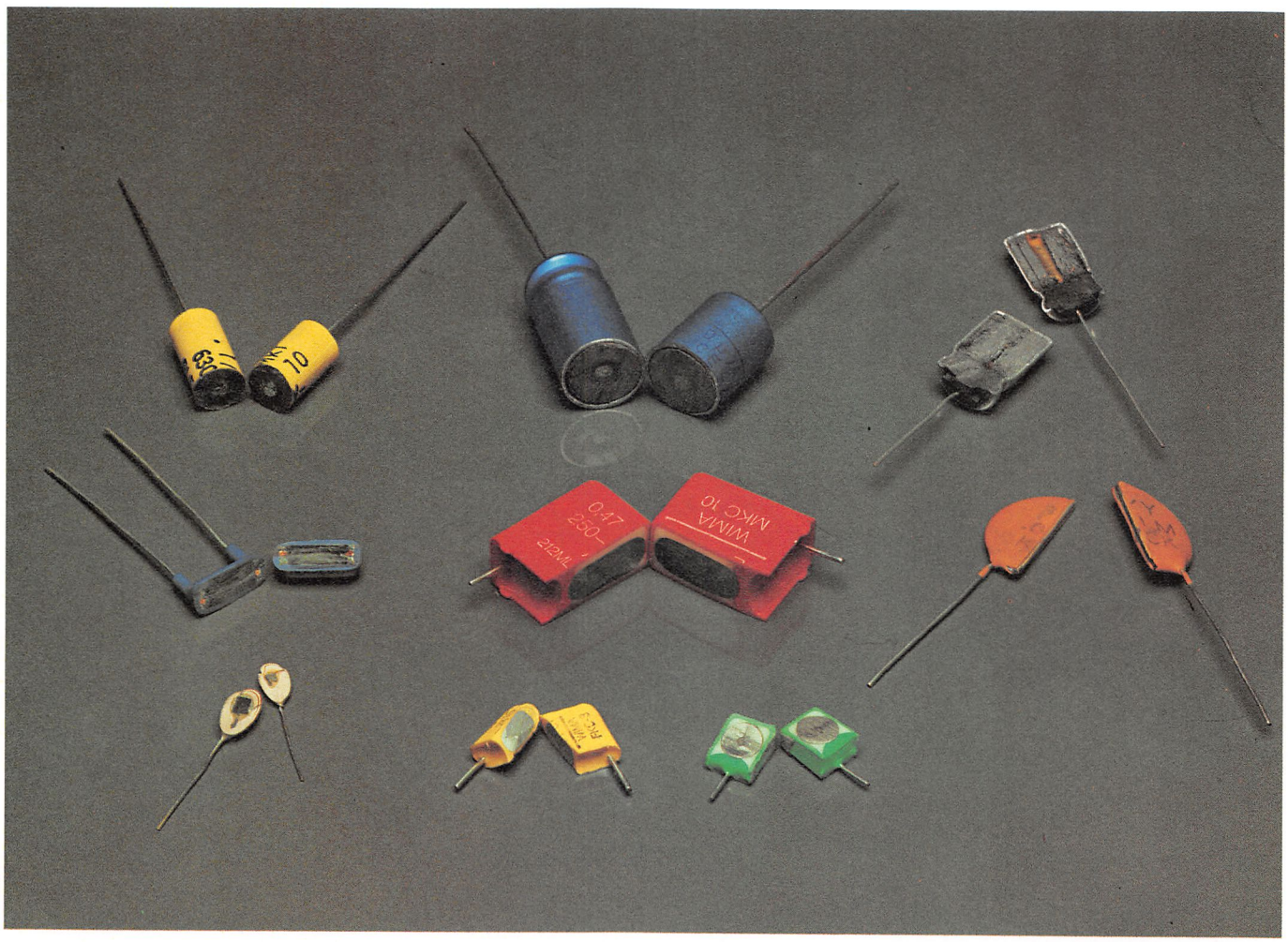
Metallic paper capacitors are used with direct and alternating voltages. The dielectric of this capacitor is made up of tiny strips of paper, one side of which has had a very fine layer of vaporized metal applied, in a vacuum. Two of these strips of metallic paper are then wrapped into cylinders with one slightly shifted out of line with the other. A metal 'bridge' is then attached to each end of the cylinder, so that the metallic strips are connected to the terminals of the component. Metallic paper capacitors have the ability to 'heal' themselves if the dielectric is perforated by high voltage. This is because the electrical arc, formed when the perforation occurs, vaporizes the neighbouring metallic strips, so that no channel is created which would allow current to pass between the plates. These capacitors are used to connect power sources of different levels and for the starting and running of motors.

Capacitors with plastic dielectrics have similar properties and uses. They are usually distinguished by the abbreviation MKV and have a dielectric made of a special sheet of polypropylene. They are produced by rolling this material with strips of metallised paper. This cylinder is put into a metal container and is impregnated with oil, in a vacuum.

Table 2 summarizes some of the characteristics of different types of capacitor. □

Table 2

Type of capacitor	Range of values	Maximum discharge voltage	Range of operating temperatures (°C)
Mica	1 pF-1 μF	50-100,000	−55 to +150
Ceramic	1 pF-2.5 μF	20-200	−55 to +125
Electrolytic	0.5-1 μF	2.5-700	−80 to +125
Paper	0.001-200 μF	50-2,000,000	−55 to +125
Trimmer	max 150 pF		
Variable	50-1000 pF		



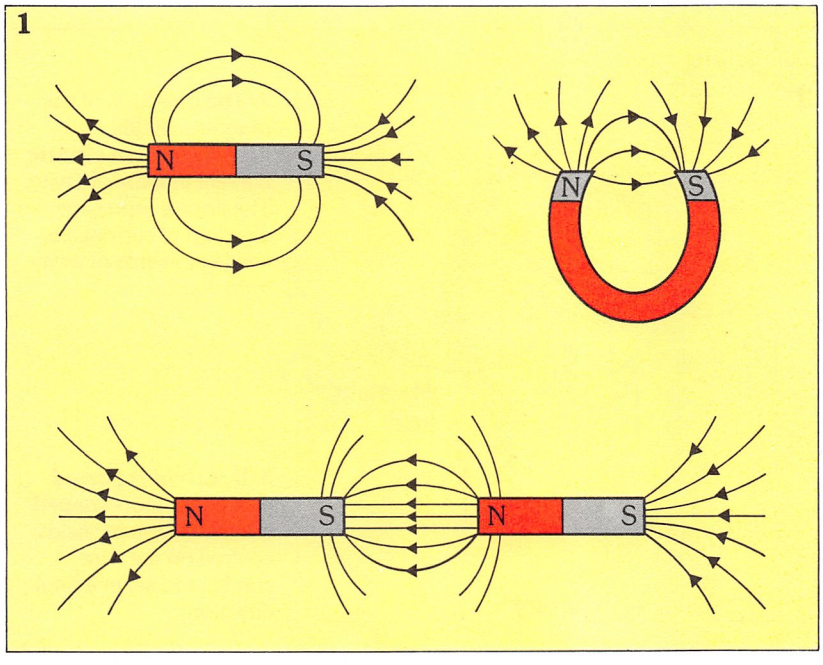
A variety of capacitors. The cross-sections clearly show the capacitive plates and the dielectric materials.

ELECTRICAL TECHNOLOGY

Magnetism and electromagnetism

We have all come across the simple magnet which attracts pieces of metal, and which (supported on a pivot) acts as a compass. As long ago as the 11th century the Chinese were familiar with the properties of magnetism in the form of the naturally occurring magnetite.

However, it wasn't till the 17th century that people realized these properties were also possessed by other substances such as iron, nickel and cobalt. In fact many substances show some magnetic properties, although in most materials the magnetism is so small as to have little practical value.



1. Typical magnetic field patterns for a bar magnet and a horse-shoe magnet.

The earliest known magnetic material was 'lode stone' or magnetite, a form of iron oxide which has been magnetized naturally by lying in the magnetic field of the earth. While iron and its compounds remain the most usual choice for magnetic material, artificial magnets are also constructed from a variety of alloys. Materials capable of being strongly magnetized are called **ferromagnetic**.

These artificially induced magnets can be divided broadly into hard and soft magnetic materials. **Hard magnetic** materials are those which are more difficult to magnetize inititially but which retain their magnetism strongly. **Soft magnetic** materials are very easily magnetized but they also lose their magnetic properties easily.

The distinction between hard and soft is not clear cut and there is a continuous range of magnetic types between them. Hard magnetic materials are used in compass needles, the armatures of electric clocks and other permanent magnets. The main application of soft materials is in transformers and electric motors, where the direction of magnetism is reversed many times per second.

Magnetic fields

To understand how some materials become permanently magnetised, we need to know more about the behaviour of electromagnetic fields.

Many of the properties of magnets may be compared with those of electrostatically charged bodies which were explained in the basic theory article on *Electrical Fields*. There are two specific forms of electric charge, negative (presence of electrons) and positive (a convenient way of describing a deficiency of electrons). Similarly in magnetism there are two types of magnetic pole, **north** (**N**) and **south** (**S**). There is an essential difference, however: while it is possible to obtain a piece of material carrying only a positive charge, it is not possible ever to obtain material carrying only a N pole or a S pole. If there is a N pole on a piece of iron then there is also a S pole of identical strength somewhere else on the same piece. Even if the piece of iron is cut in two, each resulting piece will have a pair of poles.

Let's look at the other similarities between magnets and electrical charges. Suppose a magnet is suspended by a thread at its centre. If the N pole of a second magnet is brought near the N pole of the suspended magnet, they will repel each other and the suspended magnet will tend to swing round. On the other hand if you bring the S pole of the second magnet towards the N pole of the suspended magnet they will attract each other. So similar poles repel each other while opposite poles attract. This is identical to the rule for the attraction and repulsion of electric charges.

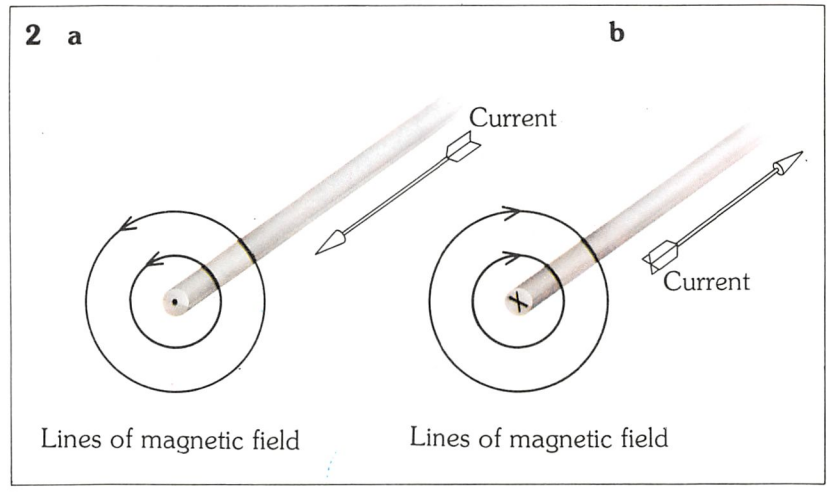
We have seen how an electric charge sets up an electric field in the surrounding space. In the area surrounding a magnet is a similar force, known as the **magnetic** or **electro-**

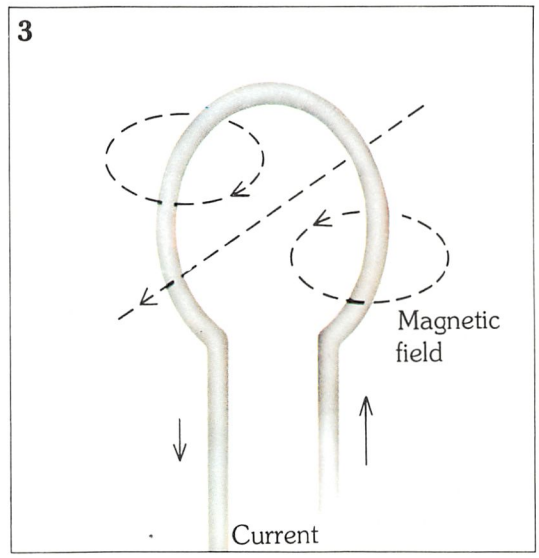
magnetic field. Diagrammatically this may be represented by lines of magnetic force starting from a N pole and ending on a S pole. To indicate a strong force these lines are drawn close together and where they are more widely spaced the force is weaker. *Figure 1* shows some typical magnetic field patterns for a bar magnet and a horse-shoe magnet. The directions of the arrows in these diagrams indicate the direction that the N pole of a compass would point when placed in the field. When two magnets are placed near to each other we notice that the two fields mix, giving the overall pattern shown.

It is very simple to demonstrate these fields by placing a bar magnet under a piece of paper and sprinkling iron filings on top. The filings within the magnetic field become magnetized, acting like tiny bar magnets, and lie along the lines of the field as shown in the photograph.

Electromagnets and solenoids

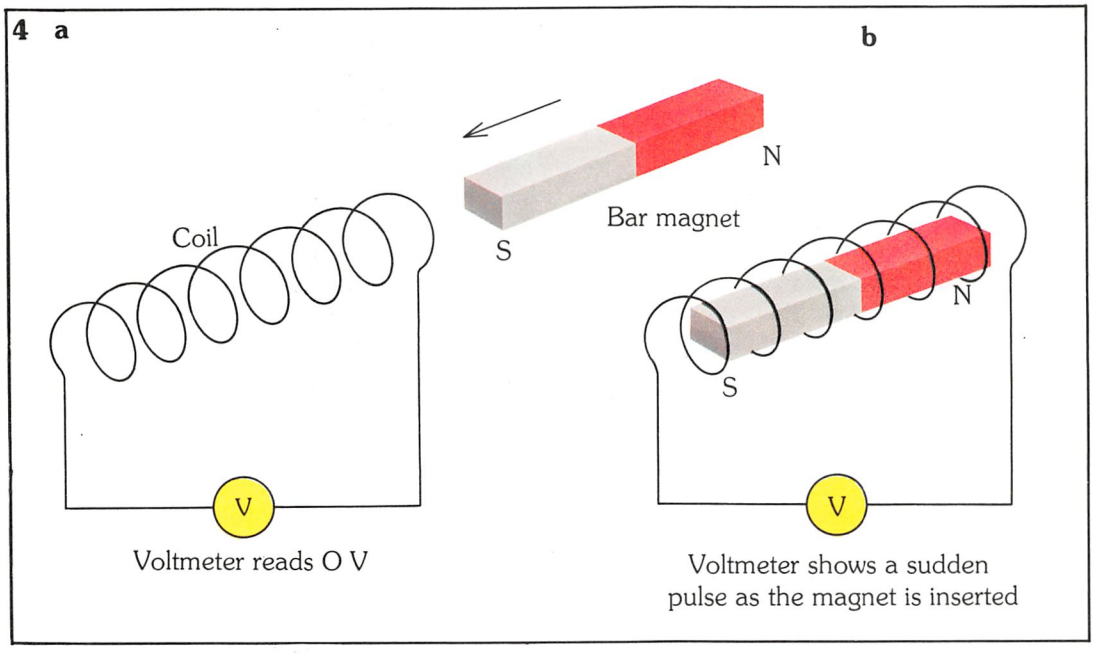
In about 1820 the Danish scientist Oersted discovered that a conductor carrying an electric current produced an electromagnetic field identical to that produced by a bar magnet. That is, it was shown to have the same effect on magnetic material that a bar magnet has. *Figure 2* illustrates the direction of the magnetic fields produced by an electric current. Using an arrow to represent direction of current, the dot in the centre of *2a* represents the point of the arrow, and the cross in *2b* represents the tail end. Oersted showed that a long straight wire carrying current produced a magnetic field moving in circles centred on the wire. The





2. The direction of the magnetic fields produced by an electric current in a straight wire. The arrows represent direction of current, i.e. flowing towards or away from you.

3. If current is passed through a loop-shaped wire the magnetic fields centred on the wire combine to act in a single direction.



4. Illustrating magnetic induction. A bar magnet inserted into a coil of wire creates a measurable voltage across the ends of the wire.

direction of the movement is clockwise when current is travelling away from you, and anti-clockwise when travelling in the opposite direction. An easy way to remember this is to think about the use of a corkscrew to remove a cork from a bottle. As you screw in the corkscrew in a *clockwise* direction, it moves *away* from you into the cork.

Figure 3 shows what happens to the magnetic field if the wire is bent into a loop. This creates a strong magnetic field because the circular fields centred on the wire are now combined to act in a single direction. We can increase the strength of the magnetic field further by passing the current round several turns of wire, which form a long coil called a **solenoid.** The magnetic field around the solenoid has a pattern exactly like that of a bar magnet.

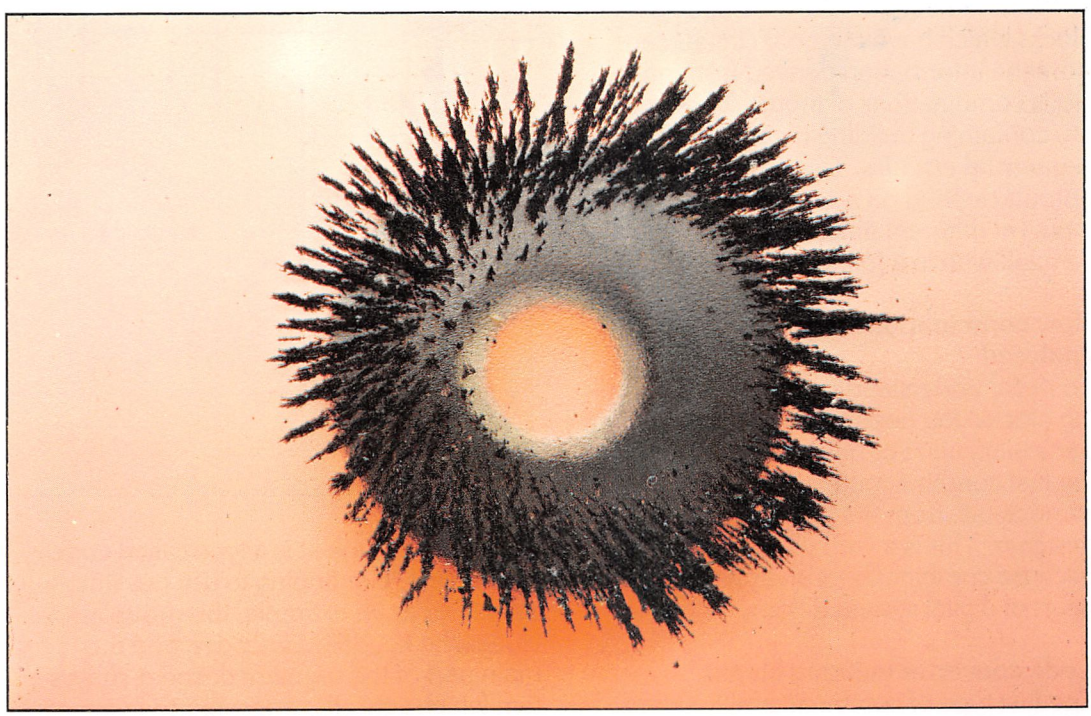
How permanent magnets are made

In every atom electrons are rotating around the nucleus, setting up an electromagnetic field perpendicular to the orbit of rotation. In effect each atom is itself an elementary magnet. However, in most materials the atoms are somewhat randomly orientated, so the individual magnetic fields cancel each other out. However in ferromagnetic materials, the atoms can be forced to align themselves parallel to each other so that the magnetic fields are all going in the same direction, with a resulting magnetic effect.

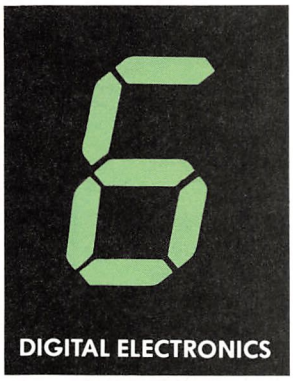
This alignment usually takes place in regions of the material called **domains.** These domains are relatively small compared with the magnet as a whole – in fact they are invisible to the naked *eye* – but each contains a large number of atoms. When ferromagnetic materials are in an unmagnetized state the domains are randomly aligned and so cancel each other out, but when magnetized they are all lined up parallel to each other. This explains why, when a magnet is broken in two each piece will have both a N and S pole.

We have seen that a current flowing in a wire gives rise to a magnetic field. A reverse effect can be obtained. If a coil of wire is placed in a magnetic field, a sudden voltage is produced across the end of the coil.

This effect is termed **magnetic induction** and can be illustrated by suddenly inserting a bar magnet into a coil of wire as shown in *figure 4*. As the magnet is pushed into the coil, the magnetic field that is created around the coil generates a voltage pulse which can be measured across the end of the coil. Where magnetic induction is used in motors, dynamos and transformers, this is usually accomplished by a change of direction of current flow in one coil of wire inducing a magnetic state and, hence, a voltage across a second adjacent coil .□



Iron filings attached to a magnetized ferrite core, illustrating the lines of the magnetic field surrounding the ferrite.



Linking digital building blocks

Combinational circuits

In earlier chapters we looked at the basic design and operation of logic gates. These gates, AND, OR, NAND, NOR and NOT, are the fundamental building blocks of digital systems. The art of digital system design is to link these building blocks together in the most effective manner. To do this, system designers use a set of logical principles to work out the most practical circuit for a particular job.

This section concentrates on understanding units which do not contain a memory. The decisions which they take depend on the input they are receiving *at that particular moment* (see *figure 1*). These are called **combinational circuits** because the output of a circuit at any given moment depends on the **combination** of input signals present. The output of a circuit is always the same for a particular input combination.

On the other hand, the building blocks which have memory circuits can store the information derived from previous combinations of inputs. These memory containing building blocks are called **sequential circuits.** This is because the outputs depend on a sequence, or chain of inputs at different times. These systems will be dealt with in a following chapter.

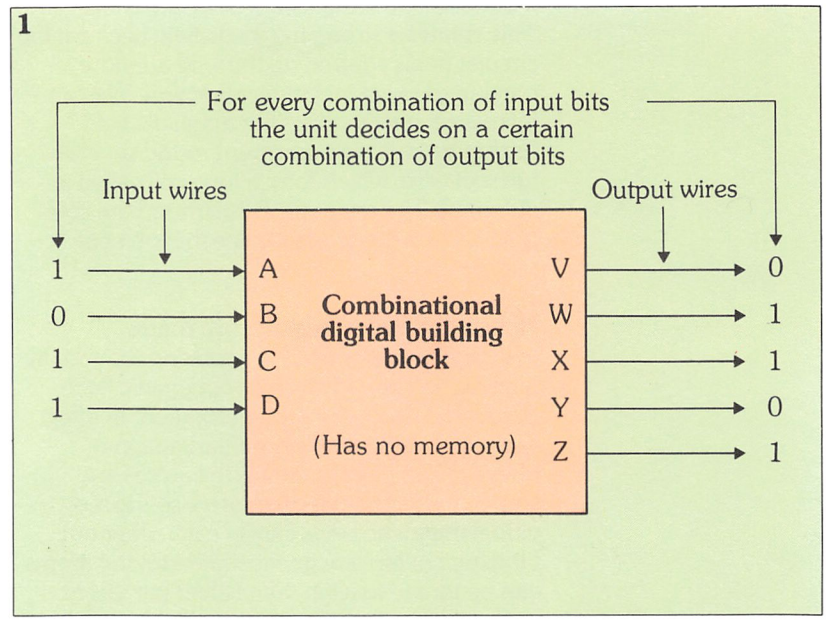
How combinational circuits are analysed

To really understand combinational building blocks you need to be able to *analyse* a particular circuit or network. We'll examine the first combinational building block in some detail, from the point of view of a designer. This is a very valuable exercise because combinational analysis is at the heart of digital system design.

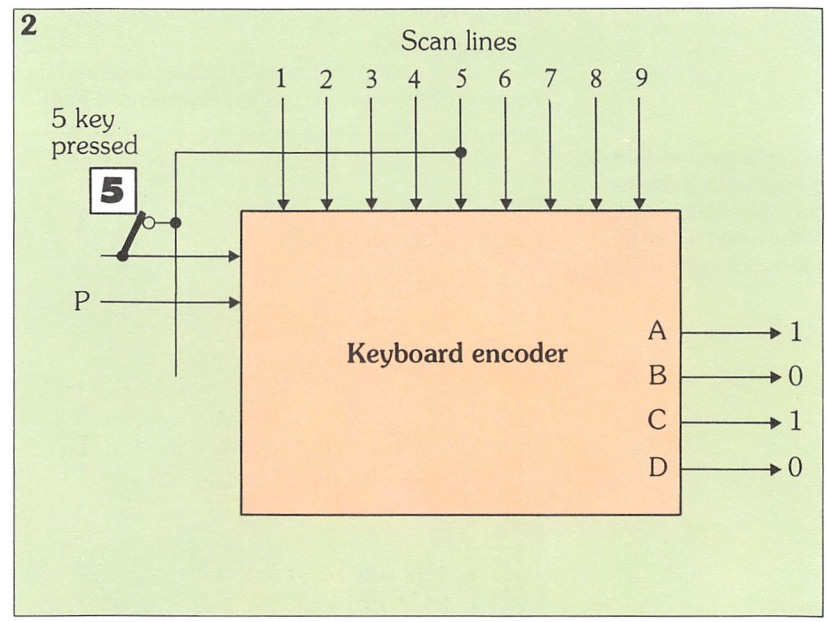
Code converter building blocks

The first type of combinational building block we will look at is a type called **code converters**. Continuing to use our simple calculator as an example, this group includes the keyboard encoder, and the main part of the segment decoder subsystem which lights the appropriate segments

1



2



1. Diagram illustrating the definition of a 'combinational' digital building block.

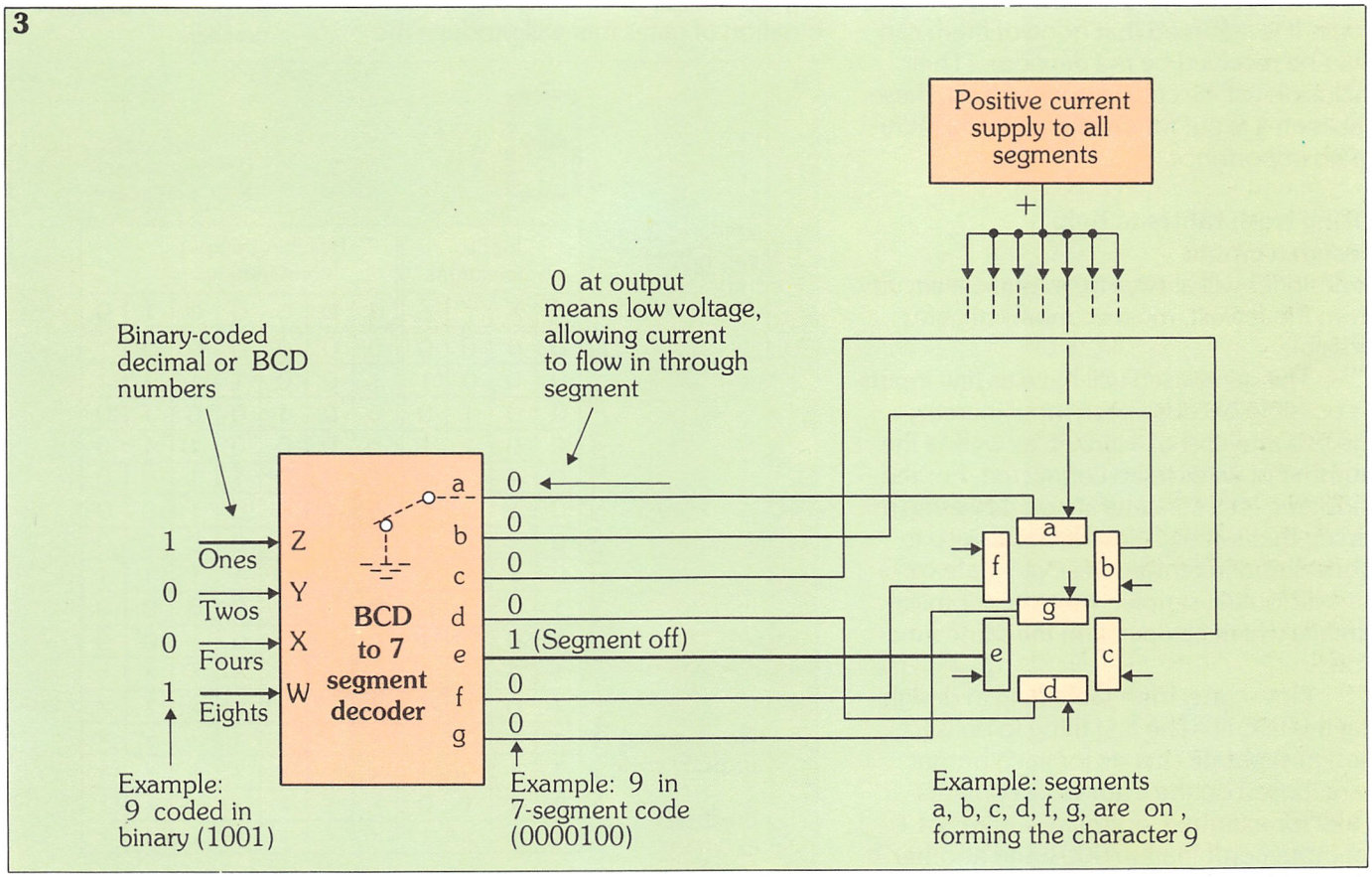
2. The keyboard encoder of a pocket calculator.

of the seven-segment LED display.

A code converter simply converts digital information from one kind of code into another. For example, in the keyboard encoder in *figure 2*, when key 5 is pressed, the input code is 10000010000. Here, the two most significant bits refer to lines N and P and the remaining nine to the scan lines.

number 9 is being received. The seven outputs labelled **a** to **g** light up a seven-segment display which shows the number being received.

The second terminal of each segment is connected to a source of current such as a scan line in the calculator. To turn a segment on, the decoder transmits a 0 in



3. Block diagram showing the function of a BCD-to-7 segment decoder. The 0 and 1 meanings are assigned using positive logic.

The corresponding output code is 0101. Code converters are a useful place to start, because their gate arrangements are a simple, logical kind found in nearly every combinational building block. If you understand this type of device, it will be much easier to analyse other types of circuit.

BCD to seven-segment decoder

We will now look at the BCD to seven-segment decoder. This type of code converter forms the main working part of the display subsystem. *Figure 3* shows what its job is.

This combinational unit receives binary coded decimal (BCD) digits at inputs W, X, Y and Z. In this example, the

that output.

For simplicity's sake, the decimal point will be left out for the moment. It can be seen that 9 corresponds to the output code 0000100. This code turns on all the segments except **e**. (Note that the logic states 1 and 0 in this electrical circuit use positive logic. High voltage means 1 and low voltage means 0.)

Designing combinational circuits

Designing any combinational building block begins with making out a truth table for it (these were discussed in *Digital Electronics 3)*. This truth table should show all the possible combinations of the various inputs and their corresponding outputs. *Figure 4* shows the truth table for

a BCD to seven-segment decoder.

The number displayed is shown on the left. For example, number 0 (first row) is sensed at input as the combination 0000. Its output code is 0000001, which turns on all segments except **g**. Note that the six combinations shown at the bottom of the table represent the binary numbers from 10 to 15. These have no meaning in BCD code. It is assumed that none of them can ever be received by the decoder. (Their inclusion will be considered later.) In these cases an **x** is put for each output to signify *of no importance.*

Using truth tables to help design a circuit

A designer will always try to make a circuit from the fewest, most economical gates possible.

The gates used will have as few inputs as possible because each input increases the size and cost of a circuit, as well as the number of wires to be connected. For the moment, let's concentrate on designing a circuit that's straightforward and easy to understand in terms of its logic. Later on, we will look at simpler, more economical circuits which can perform the same function.

How can a truth table help in designing this circuit? The first thing to do is to design separate circuits for each output wire, based on the truth table contents. Take for example, output **a**. There is a 1 for input combination 0001 and another 1 for 0100. What is the relative logic which will lead to a gate circuit for output **a**?

How do you pick the 'logic' out of a truth table?

Here's the logic behind output **a**. Decoder output **a** is 1 when we receive combination 0001, OR combination 0100. Furthermore, we have combination 0001 when W is NOT 1, AND X is NOT 1, AND Y is NOT 1, AND Z is 1. And we have combination 0100 when W is NOT 1, AND X is 1, AND Y is NOT 1, AND Z is NOT 1.

The words OR, AND and NOT in these logical statements tell us how to connect an OR gate, some AND gates, and some inverters, to make a circuit for producing output **a**. (We'll see how to recognize this circuit from the statements in a moment.)

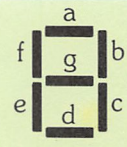
We can make similar logical statements for each of the other outputs from this building block, or from *any* combinational circuit. Statements like these can be drawn from any truth table – statements in terms of the three basic logical operations of AND, OR and NOT. And for each logical statement like this, there is a combination of gates that will produce the desired output from the given inputs. It may not be the simplest combination or most economical combination – but it will work.

So the next step in our design process, after writing the truth table, is to understand it in terms of this kind of statement.

However, logical statements like this are very cumbersome to write in plain language, they also make it difficult to see how the gates are to be connected. So digital designers use a shorthand writing for logical statements, which makes the relationship between the statements and a gate circuit very easy to see.

4. The first step in designing or understanding a combinational circuit is to make out a truth table. This is the truth table for the BCD-to-7 segment decoder.

4



1 = high voltage
0 = low voltage
X = 'don't care'

Meanings of combinations

	Input combinations				Resulting output combinations						
	W	X	Y	Z	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0
No meaning in BCD code	1	0	1	0	x	x	x	x	x	x	x
	1	0	1	1	x	x	x	x	x	x	x
	1	1	0	0	x	x	x	x	x	x	x
	1	1	0	1	x	x	x	x	x	x	x
	1	1	1	0	x	x	x	x	x	x	x
	1	1	1	1	x	x	x	x	x	x	x

Positions in binary numbers → (8 4 2 1)

The shorthand for writing logical statements

This logical shorthand, **Boolean algebra**, was developed by an Englishman named George Boole, long before digital electronics was ever thought of. Its purpose was to provide a clear and simple way of expressing complicated combinations of logical statements, which are defined as statements which are either true or false.

5. Truth table for the two-input NOR function.

5

Truth table	
A B	Y
0 0	1
0 1	0
1 0	0
1 1	0

Logic is a branch of philosophy involving the study of statements, judged to be true or false on the basis of the truth or falsity of other related statements. Boolean algebra, therefore, was a readymade method for handling the logical statements involved in every binary digital system. You may remember that in previous chapters, 1 was called true and 0 false. This is why digital gates are often called logic gates. A circuit of gates is sometimes described as a **logic circuit** or a **Boolean circuit**.

The principles of Boolean algebra

As we know, Boolean algebra considers the existence of only two distinct values. These are true (1) and false (0). They do not represent numbers or definite quantities, but rather states or characteristic conditions.

Boolean algebra was first applied to the analysis of switching circuits in 1938. Since then, it has played an important role in the design of logic circuits, particularly in the electronic digital circuits used in computers.

The logic functions of Boolean algebra will be used here as a means of developing the design of logic circuits using elements from the TTL 74 series.

Binary variables and functions

In any logical statement expressed in algebraic terms there are two elements, variables and functions. Before illustrating the concepts of Boolean algebra these terms must be defined.

Variables are the elements which have some operation carried out upon them. They are represented by symbols: A, B, C etc. The operation that is carried out upon these variables is known as a **function.** (Remember, we are dealing with binary expressions so variables can only have the values 1 or 0, and the functions are the operations AND, OR, NAND, NOR and NOT.)

If we write Y = A . B, then Y, A and B are variables. In this case Y is an **output** variable and A and B are **input** variables. The value of Y depends on the values of A and B. The AND operation that is carried out on A and B is the function.

Given, for example, the variables A and B, the function of these variables is described as Y = f (A , B), where f represents a particular relationship between A and B. For example:

Y = A + B, Y = A . B, Y = A . (A + B)

Y is a function of A and B. Likewise, in the case of three or more variables, this could happen:

Y = f (A , B , C) = A + B + C

and so on.

Given the values of A, B, etc., a value for Y (which is a function of them) is obtained through the relationship (f) between these variables. Since, in this case, Y can only be 0 or 1, it is a binary function.

A function of a real variable is normally shown in a graph. In the case of a function with binary variables, a truth table is used. In every line of this, one of the combinations of the different variables and the resulting value of the function are shown. For a function of n variables there are 2^n possible combinations, which give a truth table with 2^n lines. *Figure 5* shows the truth table for a binary function with two variables: Y = f (A , B).

The simplest binary functions are two functions with two variables, described as OR and AND, and a function of one single variable – NOT. However, before looking at these functions in detail, let's look at the laws of Boolean algebra.

Logical sum and logical product

Given two variables A and B, let us assume that the variable C = A + B. This is defined as a logical sum. If A and B are two binary variables, then C is a binary variable. Remember that in the algebra of logic, the + symbol means OR.

Given the two variables A and B, the variable C can be defined as the logical product obtained from C = A . B.

OR and AND functions

At this point the binary functions OR and AND can be introduced. Given the variables A and B, the function OR (+) represents the logical sum of them so:

Y = A + B

It is now clear why this function is called OR. The sum means the union of A with B, giving the value of Y. In this case, Y equals 1 if at least one of the two variables is 1.

(1 + 2) + 3 1 and 2 are associated

1 + (2 + 3) 2 and 3 are associated

Similarly, in terms of Boolean algebra the logical sum of the variables A, B and C can be obtained in three ways:

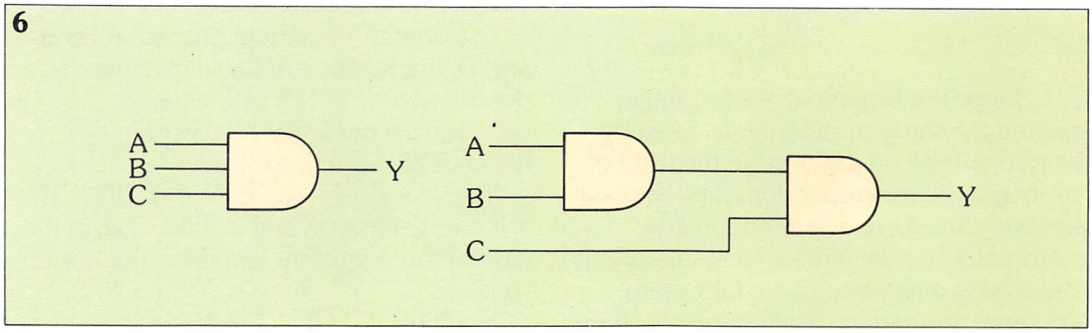
A + B + C

(A + B) + C A and B are associated

A + (B + C) B and C are associated

The principle of **association** also applies to logical products of variables, and is valid for any number of variables. The property of association has significance in terms of logic gates. *Figure 6* shows two circuits (although AND gates are used here, they could effectively be OR gates without altering the associative principle). Two, two-input AND (or OR) gates can take the place of one, three-input AND (or OR) gate, because of the property of association. The same output will be obtained because the two networks are equivalent.

6



6. Two circuits that carry out the same functions. This can be proved using Boolean algebra.

Now, what happens with the product? Given the variables A and B, the function AND (.) is defined as its logical product. Thus:

Y = A . B

In this case Y = 1 if both variables are simultaneously equal to 1. The name AND stands for something which belongs to both components of the product. So here Y will be 1 when all the variables are 1.

Understanding Boolean laws

Some of the properties of Boolean algebra used to work out logical expressions are the same as those used in ordinary mathematics: association, commutation and distribution.

In terms of ordinary maths, association means that if you add, say, the numbers 1, 2 and 3 you can do it in three ways:

1 + 2 + 3

A mathematical sum or product does not change its value when the factors are interchanged:

1 + 2 is the same as 2 + 1

3 × 4 is the same as 4 × 3

This is known as the **commutative** property. In terms of a logical sum or product:

A + B = B + A and A . B = B . A

In conventional mathematics a number which is multiplied by the sum of two or more other numbers can be written in two ways. For instance:

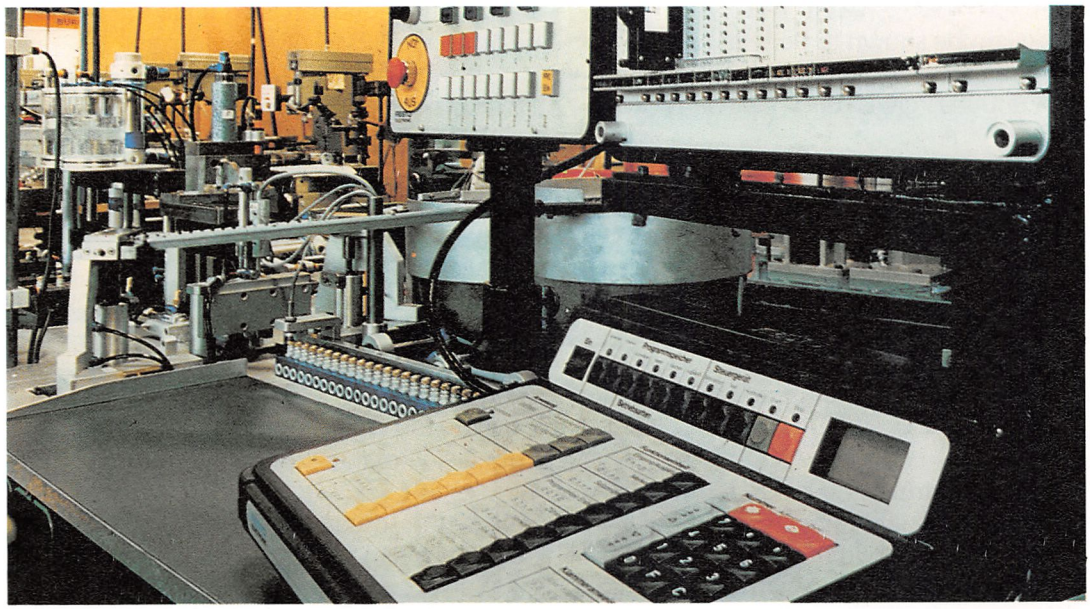
3 × (5 + 9) can be written as

(3 × 5) + (3 × 9)

The number 3 is distributed within the brackets so this property is known as **distribution.** Logical variables can be manipulated in the same way:

A . (B + C) = (A . B) + (A . C)

So A is distributed within the brackets and the logical products are obtained term by term.



Programmable controller of the type used to run manufacturing processes (Photo: Festo)

Inversion – the NOT function

Given any variable A, its complement (inverse) is defined as $\overline{A}$ (pronounced *bar A*). So, if A is 1 then its inverse $\overline{A}$ will be 0. Vice versa, if A is 0, $\overline{A}$ will be 1. The NOT binary function is therefore defined as: $Y = \overline{A}$. This function represents an inversion operation. As it has only one variable, its outcome depends on the value of A.

The principle of duality

In Boolean algebra it is sometimes possible to transform one function into another by negating its values. For instance the OR operation is the dual of the AND. Remember the truth table of a dual input AND gate:

INPUTS		OUTPUT
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Negating (inverting) all the values in this truth table gives us:

INPUTS		OUTPUT
$\overline{A}$	$\overline{B}$	X
1	1	1
1	0	1
0	1	1
0	0	0

This is the truth table for an OR gate. Similarly an OR gate can be transformed into an AND gate. The truth table for a two input OR gate is:

INPUTS		OUTPUT
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Inverting this gives us:

INPUTS		OUTPUT
$\overline{A}$	$\overline{B}$	X
1	1	1
1	0	0
0	1	0
0	0	0

Which takes us back to our original AND gate – remember an inversion of an inversion gives the original function.

A practical OR gate can be made into an AND gate (and vice versa) by placing inverters on all its inputs and its output. The duality principle is a very fundamental property of logic gates and, as we have seen, has useful practical applications.

Identity

The laws of **identity** tell us that given a variable, A, these relationships are obtained:

$A + 0 = A \qquad A \cdot 1 = A$

They can be shown to be related by applying the principle of duality: Inverting all of $A + 0 = A$ gives us:

$\overline{A} \cdot 1 = \overline{A}$

As this is a logical product, with $\overline{A}$ in both sides of the equation it can be simplified to $A \cdot 1 = A$. Quickly checking these relationships in a truth table will prove them to be correct:

A	A + 0	A . 1
0	0	0
1	1	1

Figure 7 illustrates the law of identity in terms of logic networks. Their outputs would be as given in the truth table above.

Annulment

The law of annulment says that:

$A + 1 = 1 \qquad A \cdot 0 = 0$

These can be seen to be linked by the principle of duality. Negating $A + 1 = 1$ gives us $\overline{A} \cdot 0 = 0$. Bearing in mind that this is now a logical product, and that anything multiplied by 0 = 0, we then get:

$A \cdot 0 = 0$

Complements

The law of **complements** says that:

$A + \overline{A} = 1$ and $A \cdot \overline{A} = 0$

It can then be seen that:

$A + A = A$

This is known as the law of **tautology**.

Absorption

The law of **absorption** states:

$A + (A \cdot B) = A$ and $A \cdot (A + B) = A$

The second equation can be seen to be derived from the first by again applying the principle of duality to $A + (A \cdot B) = A$. When negated this gives us:

$\overline{A} \cdot (\overline{A} + \overline{B}) = A$

It can be seen that the variable A is contained in the logical sum A + B. As the only common element is A, then the result is always dependent upon its value, i.e. it is always A, as the law of double negation tells us that $\overline{\overline{A}} = A$. It is clear that to invert an inverse will return it to its original value.

As an example of how to use these theorems and laws, we shall prove that:

$A + (\overline{A} \cdot B) = A + B$

To do this we have to show that the first side of the equation is equal to the second. The properties of distribution give us:

$(A + \overline{A}) \cdot (A + B) = A + B$

Remember, $A + \overline{A} = 1$. We now get:

$1 \cdot (A + B) = (A + B)$

By the laws of annulment we can now see that:

A + B (the left hand side) = A + B

proving the equation correct.

As another example we can show that the logical function:

$Y = (\overline{A} \cdot B \cdot C) + (\overline{A} \cdot \overline{B} \cdot C)$

can be reduced to $Y = \overline{A} \cdot C$

Taking the common factor on the right hand side of the expression gives:

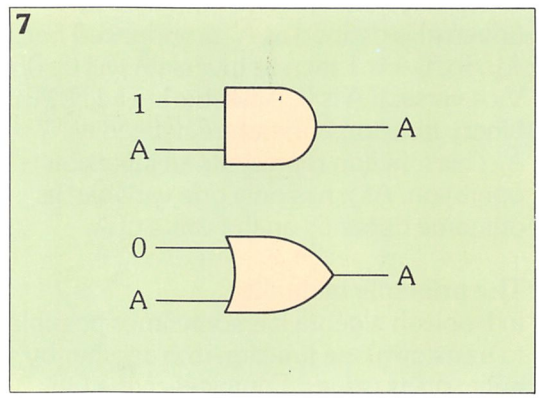
$Y = \overline{A} \cdot [(B \cdot C) + (\overline{B} \cdot C)]$

Inside the square brackets C is common, this gives us:

$Y = \overline{A} \cdot (B + \overline{B}) \cdot C$

$= \overline{A} \cdot 1 \cdot C$

$= \overline{A} \cdot C$



7. The laws of identity in logic circuits.

De Morgan's laws

There are two De Morgan's laws: the **first** states that given two variables A and B, the complement of their sum is equal to the logical product of their two variables:

$\overline{A+B} = \overline{A} \cdot \overline{B}$

The **second** law says that given the two variables A and B, the complement of their logical product is equivalent to the logical sum of their complements:

$\overline{A.B} = \overline{A} + \overline{B}$

These laws are applicable to any number of variables. Thus for four variables:

$\overline{A + B + C + D} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$

and

$\overline{A.B.C.D} = \overline{A} + \overline{B} + \overline{C} + \overline{D}$

Logical expressions

The terms product and sum were introduced into Boolean algebra to make it

easier to understand and relate to conventional mathematics. To further simplify the representation of the AND function, the dot can be omitted, and the variables related by an AND function can be written side by side: AB therefore stands for A AND B. Remember an AND function is a **logical product.** We are now writing this implicitly like the **mathematical product** of numerical variables.

A computerized test instrument used to analyse the logic functions carried out by a digital system. (Photo: Tektronix)

Summary of the laws of Boolean algebra

Absorption:
$A + (A . B) = A$
$A . (A + B) = A$

Annulment:
$A + 1 = 1$
$A . 0 = 0$

Association:
$(A + B) + C = A + (B + C)$
$(A . B) . C = A . (B . C)$

Commutation:
$A + B = B + A$
$A . B = B . A$

Complements:
$A + \overline{A} = 1$
$A . \overline{A} = 0$

De Morgan's:
$\overline{(A+B)} = \overline{A} . \overline{B}$
$\overline{(A.B)} = \overline{A} + \overline{B}$

Distributive:
$A . (B + C) = (A . B) + (A . C)$
$A + (B . C) = (A + B) . (A + C)$

Double negation:
$\overline{\overline{A}} = A$

Duality:
Whilst not an actual *law* of Boolean algebra, duality is a useful principle. An AND gate can be used as an OR and vice versa, if the inputs and output are negated. Thus:
$Y = A . B$ is the same as $\overline{Y} = \overline{A} + \overline{B}$
and:
$Y = A + B$ is the same as $\overline{Y} = \overline{A} . \overline{B}$

Identity:
$A + 0 = A$
$A . 1 = A$

Tautology:
$A . A = A$
$A + A = A$

Canonical forms

A canonical form can be a convenient way of expressing algebraic relationships. In terms of the Boolean algebra a canonical form can be used to express the relationship between the variables in a truth table. These relationships can be defined as **maxterms** and **minterms**.

A **maxterm** is obtained by taking the logical sum of the variables in a line of a truth table. If the variable is a 1, it is written as the letter of the variable which represents it (e.g. B). If it is an 0, then it is written as the inverse of the letter which represents it (e.g. $\overline{B}$).

A **minterm** is the result of the logical product of the variables in a line of a truth table. In this case, if a variable is a 1 it is written as the complement of the letter representing it (e.g. $\overline{B}$). If it is a 0, it is simply written as the letter that represents it (e.g. B).

So every line of a truth table can be represented as a minterm and a maxterm. A minterm is signified by $\mathbf{m_n}$ and a maxterm by $\mathbf{M_n}$ (where n = the number of the appropriate truth table line). Look at the truth table in *figure 8* – each line of the truth table corresponds to these minterms and maxterms:

$m_1 = \overline{A}\overline{B}C$ $\quad M_1 = A + B + \overline{C}$
$m_2 = \overline{A}B\overline{C}$ $\quad M_2 = A + \overline{B} + C$
$m_3 = \overline{A}BC$ $\quad M_3 = A + \overline{B} + \overline{C}$
$m_4 = A\overline{B}\overline{C}$ $\quad M_4 = \overline{A} + B + C$
$m_5 = A\overline{B}C$ $\quad M_5 = \overline{A} + B + \overline{C}$
$m_6 = AB\overline{C}$ $\quad M_6 = \overline{A} + \overline{B} + C$
$m_7 = ABC$ $\quad M_7 = \overline{A} + \overline{B} + \overline{C}$

Now consider all the combinations of variables which will give a result of 1 for their related logic function. One of the canonical forms for this function is given by taking the sum of all the minterms associated with the result 1. The other canonical form is obtained by taking the product of all the minterms which gives the result 0.

As an example of this, consider the truth table in *figure 9*. This has two variables, A and B, and represents an AND function. The minterms and maxterms for this truth table are:

$m_1 = \overline{A}\overline{B}$ $\quad M_1 = A + B$
$m_2 = \overline{A}B$ $\quad M_2 = A + \overline{B}$
$m_3 = A\overline{B}$ $\quad M_3 = \overline{A} + B$
$m_4 = AB$ $\quad M_4 = \overline{A} + \overline{B}$

The sum-of-products canonical form is the sum of those minterms which give a result (Y) of 1. In this case it is the single term:

$f = m_4 = AB$

8

Input combinations		
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

9

Truth table		
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

8. Input combinations for a three-input binary function.

9. Truth table for a two input AND gate.

The product-of-sum canonical form is the product of those maxterms which gave a result of 0. Here it is:

$$F = M_1M_2M_3 = (A + B)(\overline{A} + B)(A + \overline{B})$$

By the use of canonical forms, the relationships between the variables in truth tables can be expressed in an algebraic form. The sum-of-products form will tell you which combinations of variables give a result of 1 using an AND gate. The product-of-sum form on the other hand tells you which combinations of variables do not give an output of 1 using an OR gate.

In either case, to know the crucial combination of variables is enough to deduce those which bring about the opposite state.

Karnaugh maps

A Karnaugh map can be used as a more compact way of representing a truth table. As well as being more concise, Karnaugh maps also make the analysis of logic functions easier.

These maps are comprised of cells, arranged in columns and rows. An example for a two variable function is shown in *figure 10*. The columns and rows are in binary numerical (cyclic) order. Incidentally, numbering each cell from the top left hand corner across all the rows to the bottom right hand corner tells you which line of the related truth table each cell relates to. This is also illustrated in *figure 10*.

10. Karnaugh map for a two variable function.

11. Karnaugh map for a four variable function.

12. Relating the information in a truth table to a Karnaugh map.

10

A \ B	0	1
0	1	2
1	3	4

11

D C \ B A	00	01	10	11
00	0	0	0	0
01	0	0	0	0
10	0	0	0	0
11	0	0	0	1

12

Truth table for an AND gate		
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

B \ A	0	1
0	0	0
1	0	1

Truth table for an OR gate		
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

B \ A	0	1
0	0	1
1	1	1

The letters indicating the variables are shown on the top left of the map. A Karnaugh map for a function with four variables is shown in *figure 11*. In this case it represents the action of a four input AND gate – the result is 1 only when all the inputs (A,B,C and D) are 1. *Figure 12* shows the truth tables and Karnaugh maps for an OR and an AND function which have two inputs each. Although Karnaugh maps are more compact than truth tables, they can become unmanageable if they contain more than six variables.

Karnaugh maps are useful for representing the function in canonical form. Minterms and maxterms can be recognised immediately, and the variables which cause these results head the relevant columns and rows.

Reduction of logic expressions using Karnaugh maps

We have seen that canonical forms express the logic function of a system using either minterms or maxterms. In constructing logic systems we can reduce the number of gates used if we can combine a number of minterms. This is simply done by using a Karnaugh map.

Each cell in a Karnaugh map represents one minterm. If there is a 1 in two adjacent cells (for example cells 1 and 3 in *figure 10*) we can see that in this case the output is 1 for $\overline{A}\overline{B}$ and $\overline{A}B$. These differ by only one variable and may be compacted into the single term $\overline{A}$, since $\overline{A}\overline{B} + \overline{A}B = \overline{A}(\overline{B} + B) = \overline{A}$.

This shows that we may therefore link two adjacent cells both containing 1 and label this pair of cells by the logic variable common to the column or row in which they are located. Even more complex groups of cells may be compacted in systems with more than two input variables.

Putting Boolean algebra into practice

We now know that the three fundamental logic functions of Boolean algebra are the AND, OR and NOT operations. Previous chapters have shown how TTL and CMOS ICs can be used to perform logical functions. NAND and NOR gates are the most commonly available in these series of devices, as they are the easiest and most economical to produce. NAND and NOR gates can be made to perform AND and OR functions if necessary, by placing NOT gates on their outputs. Similarly a NOT gate can be made from a NAND or NOR gate with all its inputs connected together. In this way the cheapest standard circuit elements can be used to build logic circuits.

A NAND gate can be expressed as:

$$Y = \overline{A.B}$$

Applying De Morgan's laws we can rewrite this as:

$$Y = \overline{A} + \overline{B}$$

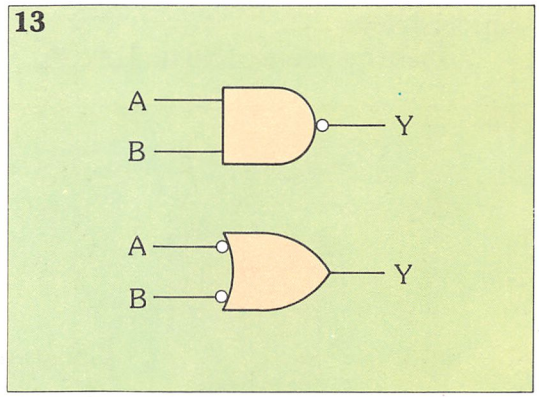
This shows that an OR gate can perform a NAND function if the input signals are inverted. A NAND gate can therefore be represented by the equivalent symbols shown in *figure 13*.

Similarly a NOR gate's output can be written as:

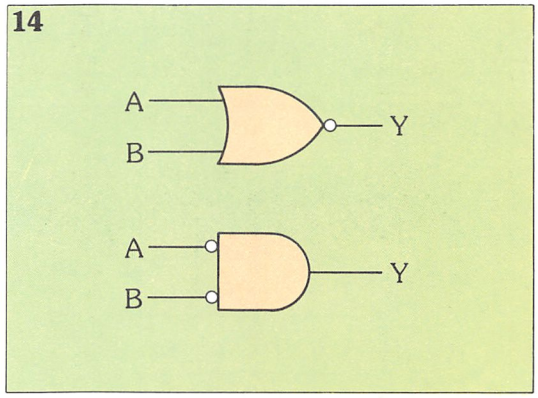
$$Y = \overline{A+B}$$

again applying De Morgan's laws we get:

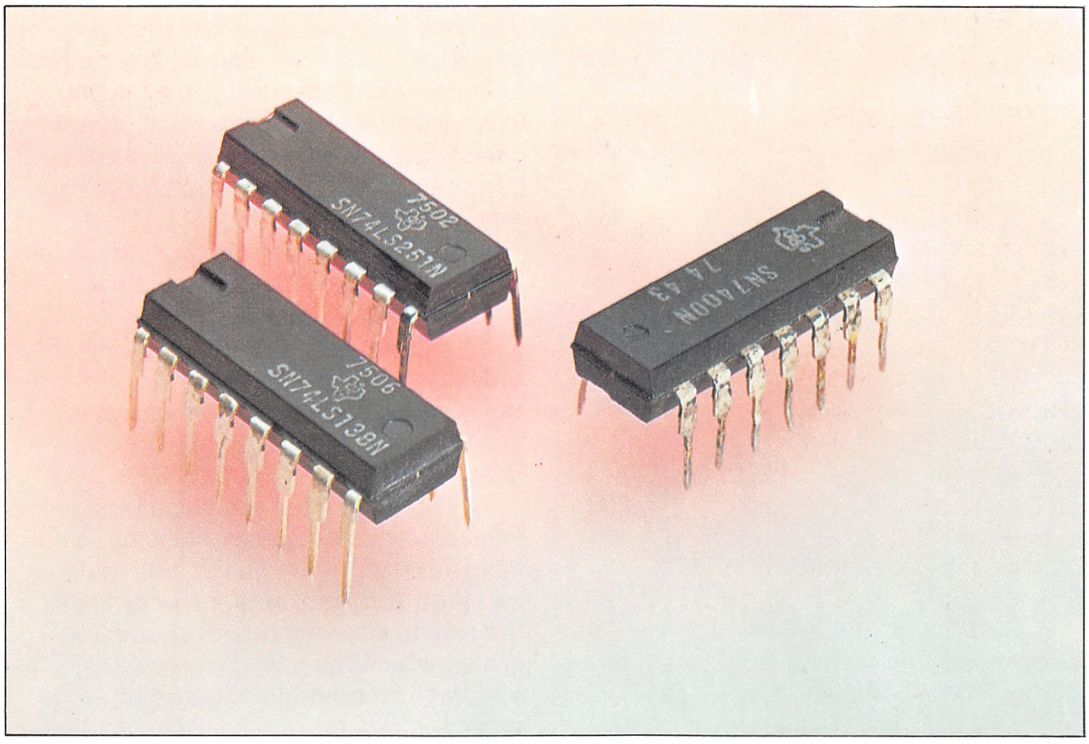
$$Y = \overline{A} . \overline{B}$$



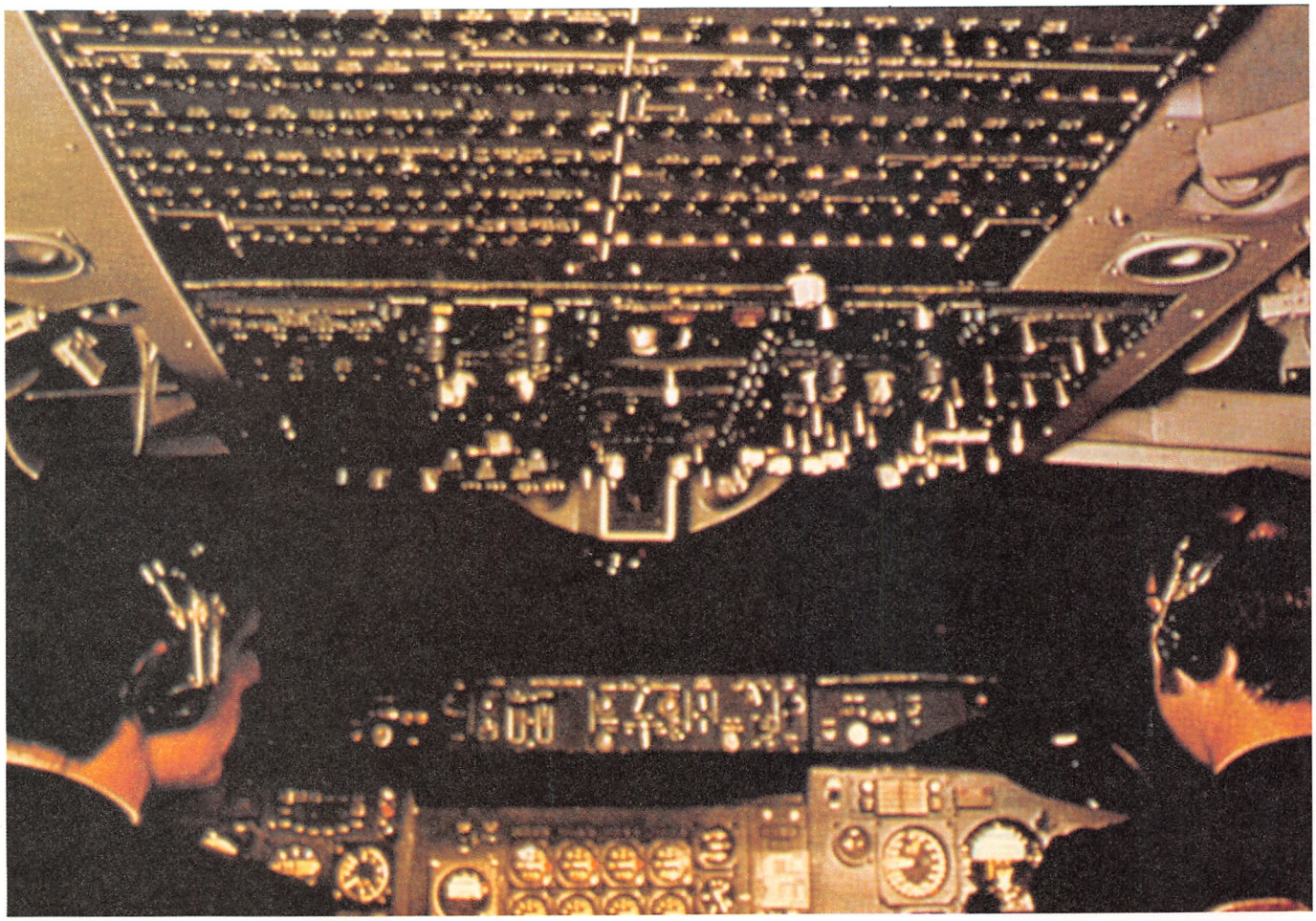
13. How the logic function of a NAND gate can be achieved by using an OR gate with inverted inputs.



14. How the logic function of an AND gate can be achieved by using a NOR gate with inverted inputs.



Integrated circuit packages that carry out combinational functions.



Digital electronics at work – the control panel on the flight-deck of a modern airliner.

This shows that a NOR gate can perform an AND function if both inputs are inverted. A NOR gate can be represented by the equivalents shown in *figure 14*.

You can now see that it is possible to obtain all the logic functions from one basic gate.

The analysis of logic circuits

According to the rules of logic, an output variable will become complementary (inverted) if it is passed through an odd number of inversion levels. It will not be complementary if the number of levels is even. The application of De Morgan's laws to NAND and NOR gates can be summarised as follows.

A NAND gate behaves like an OR gate if, keeping its inputs complemented, it is on an odd level. In the same way, a NOR gate performs an AND operation, if it is on an odd level and its inputs are complemented. Vice versa, a NAND gate performs an AND operation if it is at an even level and its inputs are not complemented.

We can now develop a set of rules which are extremely useful in interpreting the output signals of logic circuits which use NAND, and NOR and NOT gates:

1) The final gate that the output signal is taken from is defined as the first level (odd). The second level (even) is given by the gates immediately before the final stage and so on. This is shown in *figure 15*.
2) NAND gates on odd levels perform OR functions.
3) NAND gates on even levels act as AND gates.
4) NOR gates at odd levels behave as AND gates.
5) All NOR gates on even levels act as OR gates.
6) All variables, which enter a logic gate on an odd level, appear complemented at that gate's output.
7) All variables entering a gate at an even level are not complemented in output.

As an example of using these rules, look at *figure 16*. This circuit shows the various levels and their nature as a function of the desired outputs (Y_1 and Y_2).

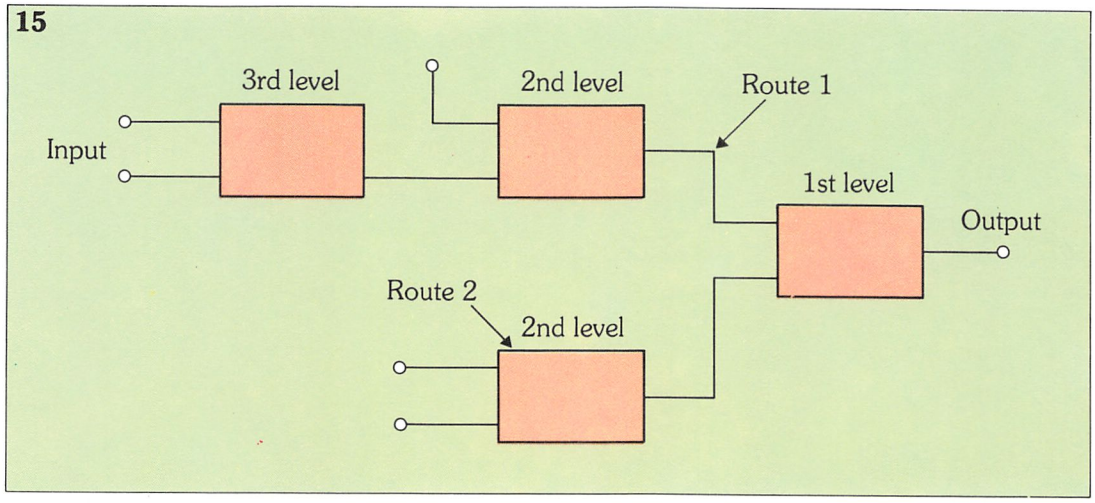
For **output Y_1** the various inversion levels which relate to this output are. defined in the following way:

a) The NAND gate 5, is placed at the first level in relation to the output, so behaves as an OR gate.

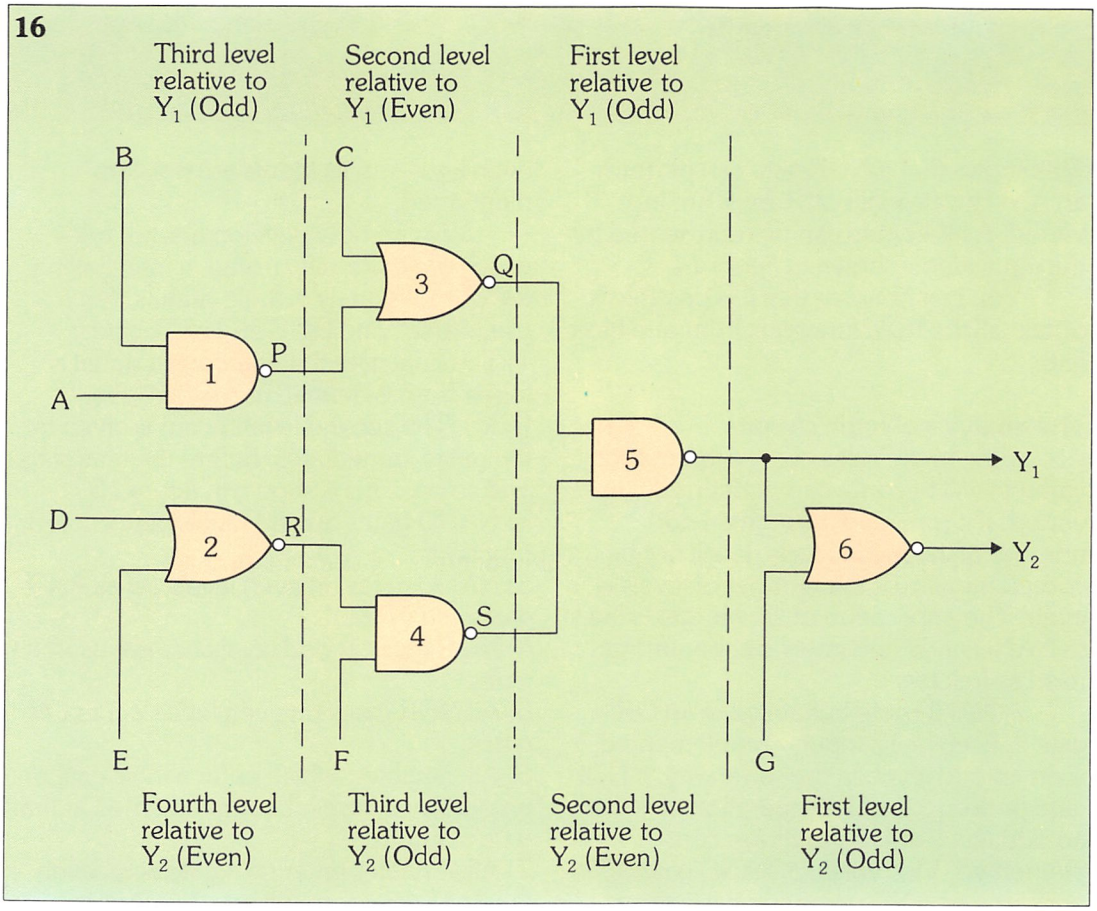
b) The NAND gate 4, on the other hand is operating on an even level, so it performs an AND operation. In the final expression of the output variable Y_1, the input F is not complemented.

c) The second level NOR gate 3 performs an OR operation. The variable C is not complemented in the final expression.

d) NOR gate 2 is on the third level. As it is on an odd level it will perform an AND



15. Level definitions in a combinational network.



16. Combinational network that gives the functions:
$Y_1 = \overline{A} + \overline{B} + \overline{C} + \overline{DEF}$
$Y_2 = AB\overline{CG}\ (D + E + \overline{F})$

operation. The input variables D and E appear complemented in the expression of Y_1.
e) NAND gate 1 is on the third level, so it acts as an OR gate. The input variables A and B will be complemented in the final expression.

To summarize these statements the final equation for Y_1 can be written as follows:

$$Y_1 = [F \,.\, (\overline{D} \,.\, \overline{E})] + [C + (\overline{A} + \overline{B})]$$

For **output Y_2** the inversion levels relate to this output in the following way:
a) NOR gate 6 is on the first level – variable G must be complemented.
b) NAND gate 5 is on an even level – the second. It performs an AND operation.
c) NAND gate 4 is on an odd level, as such it acts as an OR gate. The variable F will be complemented in the final expression for Y_2.
d) NOR gate 3 is also on the third level and will perform an AND operation. Variable C will be complemented.
e) NOR gate 2 is on the fourth level – even. It will carry out an OR operation and the variables C and D will not be complemented.
f) NAND gate 1 is also on the fourth level. It will act as an AND, and variables A and B will not be complemented.

The final expression for output Y_2 is:

$$Y_2 = \overline{G} \,.\, \{[\overline{F} + (D + E)] \,.\, [\overline{C} \,.\, (A \,.\, B]\}$$
$$= AB\overline{C}\overline{G}\,(D + E + \overline{F})$$

The final expression for Y_2 can be given as follows:

$Y_2 = Y_1 \,.\, G$, where
$\overline{Y}_1 = AB\overline{C}\,(D + E + \overline{F})$ and
$Y_2 = ABC\overline{G}\,(D + E + \overline{F})$

Glossary

binary function	a particular relationship between two or more binary variables. For example, where Y = A + B, the OR operation is the function
binary variable	a named variable which can have one of the two values 0 or 1
Boolean algebra	the fundamental system for the design and analysis of digital systems. Originally developed by George Boole for the analysis of philosophical statements
combinational circuit	a digital circuit whose output at any time depends on the combination of signals present at its inputs
canonical form	a convenient way of expressing the relationship between the variables in a truth table. Each line can be represented as a minterm and a maxterm
Karnaugh map	a compact way of representing a truth table
logical product	in Boolean algebra the AND function of two or more variables is called a logical product
logical sum	in Boolean algebra the OR function of two or more variables is called a logical sum.
maxterm	result of the logical sum of the variables in a line of a truth table
minterm	result of the logical product of the variables in a line of a truth table

Input/output peripherals

Input/output techniques and devices

Data input and output devices or **peripherals** are clearly among the most crucial components of any computer system. They provide the link between the human element – the user or operator – and the central processing unit (CPU). In the last chapter, we looked at one important set of input/output (I/O) devices, the magnetic mass storage media. These include tapes and disks, which are now familiar to most users of business and home computers. In this chapter we will be looking at other I/O peripherals, such as punched card machines, printers and VDUs, with particular emphasis on their actual use in computer systems.

The general principles explained here apply to most computer systems, so we will use a standard arrangement as an example for each peripheral. This will have at least a status register and an I/O buffer.

The **status register** is composed of a set of 1-bit stores which supply essential information about the current status of the peripheral, for example, if paper was low on a printer, or if a buffer was occupied and the data had to wait to be handled.

The **I/O buffer** is a transit register. The size of the buffer varies depending on the type of peripheral it is serving. It stores data temporarily while information is being transferred to and from the CPU to another part of the system – in this case a peripheral. In addition to this, an I/O operation necessitates an I/O buffer actually within the central memory of the computer to take the data to or from a peripheral's I/O buffer.

Punched card devices

Although punched cards and tapes are not now used in computing as often as they used to be, they are worth looking at to help build a picture of the development of I/O techniques.

If data or instructions are to be supplied to a computer from punched cards, the following equipment is needed:
– a punching machine to transform the original information into punched card form;
– a verifying machine to check that the punching of the documents is correct;
– a selecting machine to ensure the punched cards are in the right order (in fact, this kind of machine is rarely used today, as 'selection programs' can be used to make sure the cards are entered in order);
– a card reader connected to the computer, to feed in the data on the cards.

1. Card punching machine. (Photo: Honeywell)

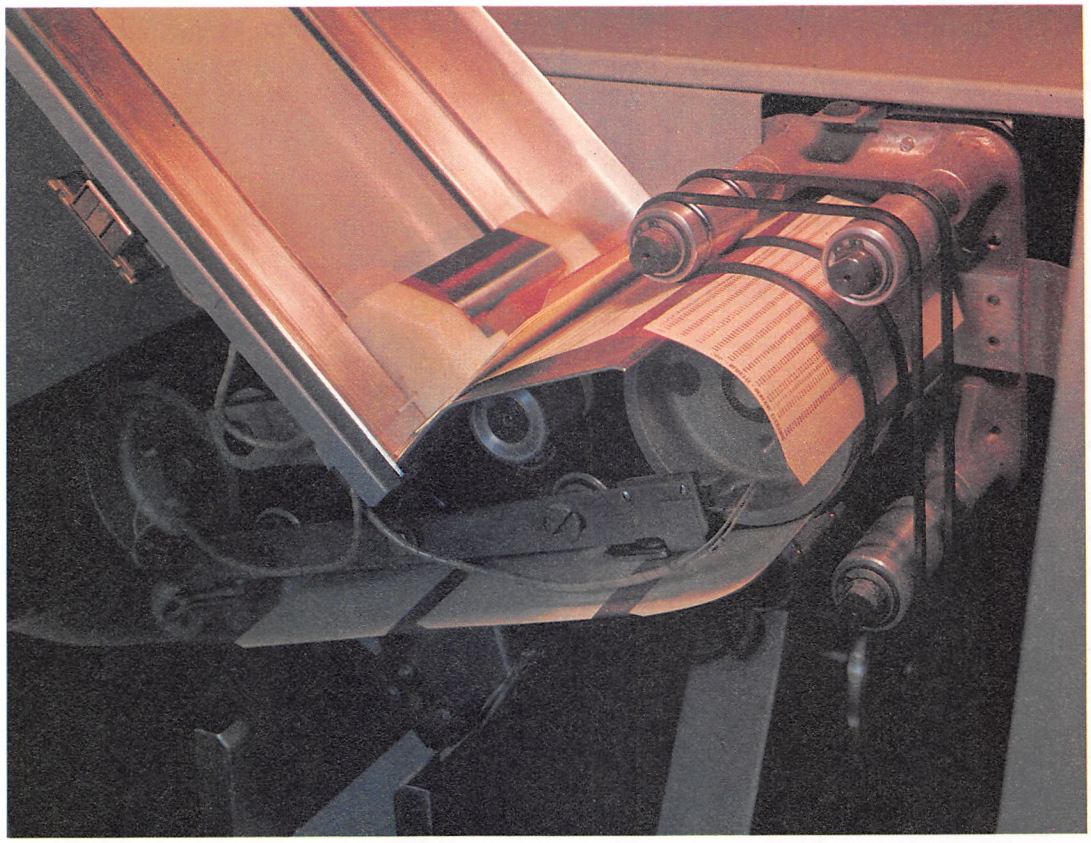
The card itself is simply a rectangular piece of card, divided into 80 columns and

12 channels (rows). These channels are numbered from 0 to 9, with two channels above the 0 channel (11 and 12).

The way this card works is to represent the data, character by character, as combinations of holes. The specific combinations used are defined by a special code. Each card will either contain a single program instruction, or an item of input data for use by a program.

position representing a 0 bit. The computer reads each instruction (i.e. one card) at a time by using its **read buffer**. It will perform the instructions only if they are interpreted according to the program directives.

A layout is established for every card before it is punched. This sets out the number of characters needed for each piece of data and in which columns of the card it must be punched. Against this, the



Right: Close-up of a card-reading machine. (Photo: Honeywell)

Figure 1 shows a Honeywell punching machine which is a machine for actually punching the holes in cards. This operation is controlled by a keyboard, like a typewriter.

A **verifier** is another keyboard operated device which checks the way data has been punched into the cards. A card reader reads the information represented by the holes and converts it into a form which is usable for processing by the computer. Reading is achieved either electromechanically or photo-electrically, and can be either serial or parallel. In either case, the final result is a photograph of the state of the holes on the card – a hole representing a 1 bit, and a non-perforated

program must be informed, by means of the conventions of the language used, as to the characteristics of the data which is in the buffer. It can then group the characters which belong to each piece of data according to the code in which they are written. It also carries out any code conversions and transfers them from the buffer in the memory zones devoted to them.

Reading a punched card is, in fact, fairly complicated. While the card moves at a constant speed under the reading station, the computer must read it column by column or line by line with the correct timing, so as not to mix up separate pieces of data.

In old machines a **collator** was used

alongside the card selector, its job being to merge two packs of cards which had already been sorted into a single sequence. Later computers are able to do this job on magnetic devices utilising a **merger program** (which can sort two or more sets of records to create a single file).

Punched paper tape devices

Like cards, punched paper tape has been used in computers since the early days. Today it is usually only used in machine-tool control and telex applications. The tape itself is a strip of electrically resistant opaque paper or plastic with a series of sprocket holes along its length. The characters are recorded as rows of holes across the width – usually 5, 6, 7 or 8. This tape is referred to as 6-track, 7-track, etc. One bit is always used for parity control, as can be seen in *figure 2* where tracks 1 to 7 are used for the code, and 8 for the parity check.

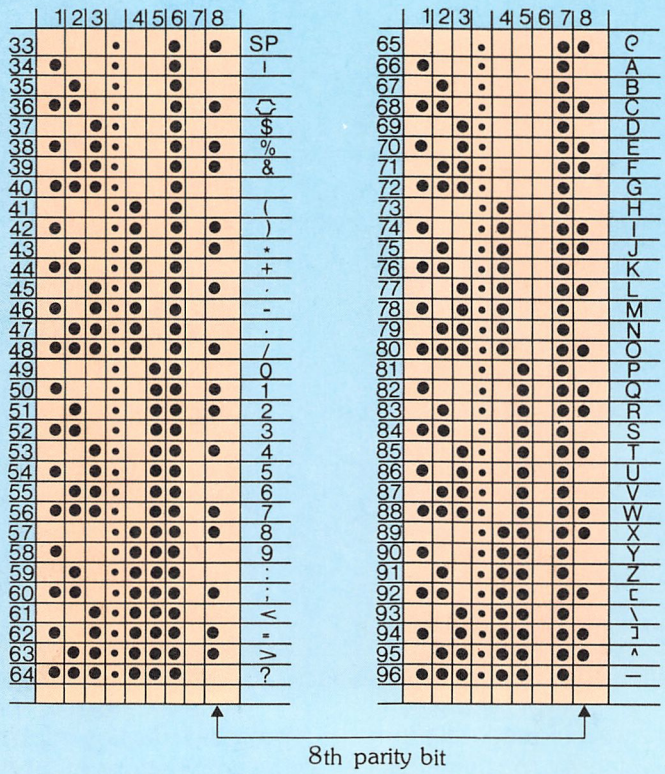
Punched tape takes up less space than cards, and can be read at speeds of up to 1,000 characters per second. However, one significant disadvantage is that one error character on a tape means complete re-punching – with cards, only the individual card with the error would need to be re-punched.

Punched paper tapes are stored on spools and fed at high speed through peripherals known as **readers.** In some cases they can be read in blocks, in others in a serial manner, just as with the magnetic tape discussed in the last chapter.

The process of punching and verification is carried out (for magnetic as well as punched media) by two keyboard operators. The first operator will transfer the data to the storage medium. When this is completed a second keyboard operator will type in all the data again, while the verifying machine compares it to the original version. If there is a difference between the work of the first and second operators an alarm will sound, and the operator will check the original version, and if neccessary make a correction.

(continued in Part 9)

2



2. Diagram of a section of punched paper tape, showing how each track can be used to represent a particular number, letter or symbol. Tracks 1 to 7 are used for code, and the 8th is for the parity check.